



UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

Grado en Ingeniería Informática

Trabajo Fin de Grado

DESARROLLO DE PLATAFORMA WEB PARA EL USO DE
HERRAMIENTAS DE SIMULACIÓN

Autor: Javier García Pérez

Tutor: Israel González Carrasco

Junio 2015

TABLA DE CONTENIDOS

1.	Abstract	1
2.	Introducción	11
2.1.	Objetivos	12
2.2.	Principales herramientas utilizadas	12
3.	Definiciones y Acrónimos	14
3.1.	Definiciones y acrónimos	14
4.	Estudio de viabilidad del sistema (EVS).....	16
4.1.	Establecimiento del alcance del sistema.....	16
4.1.1.	Estudio de la solicitud.....	16
4.1.2.	Identificación del alcance del sistema.....	17
4.1.3.	Identificación de los interesados en el sistema (Stakeholders)	18
4.2.	Estudio de la situación actual.....	19
4.2.1.	Valoración del estudio de la situación actual.....	19
4.3.	Definición de los requisitos del sistema	21
4.3.1.	Identificación de requisitos	22
4.3.1.1.	Requisitos de capacidad	23
4.3.1.2.	Requisitos de restricción	27
4.3.1.3.	Requisitos inversos.....	30
4.4.	Estudio de alternativas a la solución	31
4.4.1.	Alternativa 1: Java Servlets + MySQL	31
4.4.2.	Alternativa 2: PHP + MySQL	33
4.5.	Valoración de las alternativas	33
4.5.1.	Valoración de alternativa 1: Java Servlets + MySQL	34
4.5.2.	Valoración de alternativa 2: PHP + MySQL	34
4.6.	Selección de la solución	34
5.	Gestión del proyecto	38
5.1.	Ciclo de vida	38
5.2.	Organización del proyecto	39
5.3.	Planificación	41
5.4.	Estimación de costes	44
6.	Análisis del sistema	46

6.1.	Objetivo	46
6.2.	Alcance	46
6.3.	Definición del sistema	46
6.3.1.	Determinación del alcance del sistema.....	46
6.3.2.	Identificación del entorno tecnológico	47
6.3.3.	Especificación de estándares y normas.....	47
6.3.3.1.	Restricciones generales.....	47
6.3.3.2.	Supuestos y dependencias	48
6.3.3.3.	Entorno operacional.....	48
6.3.4.	Identificación de los usuarios participantes y finales.....	49
6.3.5.	Estudio de la seguridad requerida en el proceso de análisis del sistema de información	49
6.4.	Establecimiento de requisitos software.....	49
6.4.1.	Especificación de casos de uso.....	50
6.4.1.1.	CU-00: Acceso a la aplicación web	51
6.4.1.2.	CU-01: Cierre de sesión	52
6.4.1.3.	CU-02: Gestión de cuenta	53
6.4.1.4.	CU-03: Subir caso de simulación	54
6.4.1.5.	CU-04: Gestión de casos.....	55
6.4.1.6.	CU-05: Gestión de permisos de caso.....	56
6.4.1.7.	CU-06: Elegir idioma	57
6.4.1.8.	CU-07: Navegación entre casos de simulación	58
6.4.2.	Obtención de requisitos	58
6.4.2.1.	Requisitos funcionales.....	60
6.4.2.2.	Requisitos de calidad.....	65
6.4.2.3.	Requisitos de documentación	66
6.4.2.4.	Requisitos de seguridad	66
6.4.2.5.	Requisitos de comprobación.....	67
6.4.2.6.	Requisitos de rendimiento	69
6.4.2.7.	Requisitos de interfaz.....	71
6.4.2.8.	Requisitos de operación	72
6.4.2.9.	Matriz de trazabilidad: UR-SR	75
6.5.	Identificación de subsistema de análisis	78
6.5.1.	Determinación de subsistemas de análisis	78
6.6.	Análisis de los casos de uso.....	81
6.6.1.	Identificación de clases asociadas a un caso de uso	82

6.6.1.1.	Descripción de la iteración de objetos	82
6.7.	Análisis de clases	89
6.7.1.	Identificación de responsabilidades y atributos	91
6.7.2.	Identificación de asociaciones y agregaciones.....	96
6.7.3.	Identificación de generalizaciones	97
6.8.	Elaboración del modelo de datos.....	97
6.8.1.	Especificación de necesidades y de migración de datos y carga inicial	98
6.9.	Definición de interfaces de usuario	98
6.9.1.	Especificación de principios generales de la interfaz	99
6.9.2.	Identificación de perfiles y diálogos.....	100
6.9.3.	Especificación de formatos individuales y de la interfaz de pantalla.....	101
6.9.3.1.	UI-00: Página de inicio.....	101
6.9.3.2.	UI-01: Menú	103
6.9.3.3.	UI-02: Repositorio	103
6.9.3.4.	UI-03: Gestión de cuenta.....	104
6.9.3.5.	UI-04: Subida de casos	105
6.9.3.6.	UI-05: Panel de compartición.....	105
6.9.3.7.	UI-06: Vista de solución.....	106
6.9.3.8.	UI-07: Error/información.....	106
6.9.4.	Especificación del comportamiento dinámico de la interfaz	107
6.9.5.	Especificación de formatos de impresión	108
6.10.	Análisis de consistencia y especificación de requisitos.....	108
6.10.1.	Verificación y análisis de consistencia entre modelos	108
6.10.2.	Validación de los modelos.....	109
7.	Diseño del sistema	110
7.1.	Objetivo	110
7.2.	Alcance	110
7.3.	Definición de la arquitectura del sistema	110
7.3.1.	Definición de niveles de arquitectura	111
7.3.2.	Especificación de excepciones	112
7.3.3.	Especificación de estándares y normas de diseño y construcción.....	114
7.3.4.	Identificación de subsistemas de diseño.....	114
7.3.5.	Especificación del entorno tecnológico	115
7.3.6.	Especificación de requisitos de seguridad y operación.....	116
7.3.7.	Estudio de la seguridad requerida en el proceso del sistema de información .	116
7.3.8.	Análisis de los riesgos del entorno tecnológico	117

7.4.	Diseño de la arquitectura de soporte	117
7.4.1.	Diseño de subsistemas de soporte.....	117
7.5.	Diseño de caso de uso reales	118
7.5.1.	Identificación de clases asociadas a un caso de uso	119
7.5.2.	Diseño de la realización de los casos de uso	120
7.5.2.1.	CU-00: Acceso a la aplicación web	121
7.5.2.2.	CU-01: Usuario logueado cierra sesión	122
7.5.2.3.	CU-02: Gestión de cuenta	123
7.5.2.4.	CU-03: Subir caso de simulación	123
7.5.2.5.	CU-04: Gestión de casos.....	124
7.5.2.6.	CU-05: Gestión de permisos de caso.....	126
7.5.2.7.	CU-06: Elegir idioma	127
7.5.2.8.	CU-07: Navegación entre casos de simulación	128
7.5.3.	Revisión de la interfaz de usuario	128
7.5.3.1.	UI-00: Página de inicio	129
7.5.3.2.	UI-01: Menú	131
7.5.3.3.	UI-02: Repositorio	131
7.5.3.4.	UI-03: Gestión de cuenta.....	134
7.5.3.5.	UI-04: Subida de casos	136
7.5.3.6.	UI-05: Panel de compartición.....	137
7.5.3.7.	UI-06: Vista de solución.....	139
7.5.3.8.	UI-07: Error/Información.....	139
7.5.4.	Revisión de subsistemas de diseño e interfaces	141
7.6.	Diseño de clases	141
7.6.1.	Identificación de clases de diseño.....	141
7.6.1.1.	Capa vista	141
7.6.1.2.	Capa controlador.....	142
7.6.1.3.	Capa modelo	148
7.6.2.	Diseño de asociaciones y agregaciones	154
7.6.3.	Identificación de atributos de las clases	154
7.6.4.	Identificación de los métodos de las clases	158
7.6.5.	Diseño de jerarquías.....	173
7.6.6.	Especificación de necesidades de migración y carga inicial de datos	174
7.7.	Diseño físico de datos	174
7.7.1.	Diseño del modelo físico de datos	174

7.7.2.	Especificación de caminos de acceso a los datos y optimización del modelo físico	177
7.7.3.	Especificación de la distribución de datos	177
7.8.	Verificación y aceptación de la arquitectura del sistema	177
7.8.1.	Verificación de las especificaciones de diseño.....	178
7.8.2.	Análisis de consistencia de las especificaciones de diseño	178
7.9.	Especificación técnica del plan de pruebas.....	183
8.	Plan de validación y verificación del software	184
8.1.	Objetivo	184
8.2.	Especificación del plan de pruebas	184
8.2.1.	Definición del alcance de pruebas	184
8.2.2.	Pruebas unitarias.....	185
8.2.3.	Pruebas de aceptación	190
8.2.4.	Pruebas de sistema	193
8.2.5.	Pruebas de implantación.....	194
8.2.6.	Pruebas de integración.....	195
8.3.	Análisis de consistencia	201
9.	Conclusiones	203
9.1.	Posibles mejoras.....	203
10.	Bibliografía	205
ANEXO I.	Manual de usuario	206
1.	Acceso a la aplicación.....	206
2.	Cambiar datos de registro.....	208
3.	Subir caso de simulación	209
4.	Uso del repositorio	211
5.	Compartir caso	214
6.	Desconectar de la aplicación.....	215
7.	Cambiar idioma	215
8.	Navegación entre listado de casos.....	216
ANEXO II.	Manual de instalación	218
1.	Base de datos	218
2.	Servidor de aplicaciones.....	218
3.	Servidor de simulaciones	219
4.	Montar aplicación	220

ÍNDICE DE FIGURAS

Illustration 1 MVC - Java Servlets	4
Illustration 2 Front-Controller	5
Illustration 3 MCV + Front Controller	5
Illustration 4 database diagram UML	9
Ilustración 5 Diagrama de flujo de la App	17
Ilustración 6 Arquitectura Servlet	32
Ilustración 7 Modelo de persistencia	32
Ilustración 8 Resumen alternativa 1	32
Ilustración 9 Arquitectura PHP	33
Ilustración 10 Alternativas: Media sin ponderar	35
Ilustración 11 Alternativas: Parámetros sin ponderar	36
Ilustración 12 Alternativas: Medias ponderadas	36
Ilustración 13 Diagrama ejemplo de ciclo de vida en cascada	39
Ilustración 14 Descomposición de tareas	40
Ilustración 15 Diagrama GANTT (resumido)	41
Ilustración 16 GANTT Gestión de proyecto	41
Ilustración 17 GANTT Estudio de viabilidad	42
Ilustración 18 GANTT Análisis del sistema	42
Ilustración 19 GANTT Diseño del sistema	42
Ilustración 20 GANTT Desarrollo del sistema	42
Ilustración 21 GANTT Integración del sistema	43
Ilustración 22 GANTT Entrega	43
Ilustración 23 Estimación de recursos	44
Ilustración 24 Diagrama ejemplo caso de uso	50
Ilustración 25 CU-00	51
Ilustración 26 CU-01	52
Ilustración 27 CU-02	53
Ilustración 28 CU-03	54
Ilustración 29 CU-04	55
Ilustración 30 CU-05	56
Ilustración 31 CU-06	57
Ilustración 32 CU-07	58
Ilustración 33 MVC - Java Servlets	79
Ilustración 34 Front-Controller	79
Ilustración 35 MVC-Sistema	80
Ilustración 36 MVC-Sistema en detalle	81
Ilustración 37 Diagrama de secuencia CU-00_1	83
Ilustración 38 Diagrama de secuencia CU-00_2	83
Ilustración 39 Diagrama de secuencia CU-00_3	84
Ilustración 40 Diagrama de secuencia CU-01	84
Ilustración 41 Diagrama de secuencia CU-02	84

Ilustración 42 Diagrama de secuencia CU-03.....	85
Ilustración 43 Diagrama de secuencia CU-04_1.....	85
Ilustración 44 Diagrama de secuencia CU-04_2.....	86
Ilustración 45 Diagrama de secuencia CU-04_3.....	86
Ilustración 46 Diagrama de secuencia CU-04_4.....	87
Ilustración 47 Diagrama de secuencia CU-04_5.....	87
Ilustración 48 Diagrama de secuencia CU-5_1.....	88
Ilustración 49 Diagrama de secuencia CU-5_2.....	88
Ilustración 50 Diagrama de secuencia CU-05_3.....	88
Ilustración 51 Diagrama de secuencia CU-06.....	89
Ilustración 52 Diagrama de secuencia CU-07.....	89
Ilustración 53 Análisis clases	90
Ilustración 54 Modelo E/R de db.....	98
Ilustración 55 IU-00(1).....	102
Ilustración 56 IU-00(2).....	102
Ilustración 57 IU-00(3).....	103
Ilustración 58 IU-01	103
Ilustración 59 UI-02	104
Ilustración 60 UI-03	104
Ilustración 61 UI-04	105
Ilustración 62 UI-05	106
Ilustración 63 IU-06	106
Ilustración 64 UI-07	107
Ilustración 65 Mapa de navegación	107
Ilustración 66 Arquitectura del sistema	111
Ilustración 67 Control de versiones.....	116
Ilustración 68 CU-00_1.....	121
Ilustración 69 CU-00_2	122
Ilustración 70 CU-00_3.....	122
Ilustración 71 CU-01.....	123
Ilustración 72 CU-02.....	123
Ilustración 73 CU-03.....	124
Ilustración 74 CU-04_1.....	124
Ilustración 75 CU-04_2.....	125
Ilustración 76 CU-04_3.....	125
Ilustración 77 CU-04_4.....	125
Ilustración 78 CU-04_5.....	126
Ilustración 79 CU-05_1.....	126
Ilustración 80 CU-05_2.....	127
Ilustración 81 CU-05_3.....	127
Ilustración 82 CU-06.....	128
Ilustración 83 CU-07.....	128
Ilustración 84 UI-00_1	129
Ilustración 85 UI-00_2	130
Ilustración 86 UI-00_3	130

Ilustración 87 UI-01_2	131
Ilustración 88 UI-01_1	131
Ilustración 89 UI-02_1	132
Ilustración 90 UI-02_2	132
Ilustración 91 UI-02_3	133
Ilustración 92 UI-02_4	134
Ilustración 93 UI-03_1	135
Ilustración 94 UI-03_2	135
Ilustración 95 UI-04_1	136
Ilustración 96 UI-04_2	137
Ilustración 97 UI-05_1	138
Ilustración 98 UI-05_2	138
Ilustración 99 UI-06	139
Ilustración 100 UI-07_1	140
Ilustración 101 UI-07_2	140
Ilustración 102 ServletController.java	143
Ilustración 103 AccountServlet.java	143
Ilustración 104 ActivateServlet.java	143
Ilustración 105 DeleteServlet.java	144
Ilustración 106 DownloadServlet.java	144
Ilustración 107 LanguageServlet.java	144
Ilustración 108 LoginServlet.java	145
Ilustración 109 LogoutServlet.java	145
Ilustración 110 RunServlet.java	145
Ilustración 111 RegisterServlet.java	146
Ilustración 112 PageServlet.java	146
Ilustración 113 RecoverServlet.java	146
Ilustración 114 SeeServlet.java	147
Ilustración 115 ShareServlet	147
Ilustración 116 UploadServlet.java	147
Ilustración 117 User.java	148
Ilustración 118 Project.java	149
Ilustración 119 Sharing.java	149
Ilustración 120 Deletedproject.java	150
Ilustración 121 Satatuscategory.java	150
Ilustración 122 ProxyManager.java	150
Ilustración 123 Funciones.java	151
Ilustración 124 Comunicacion.java	152
Ilustración 125 Simulate.java	153
Ilustración 126 MessageRQ.java	153
Ilustración 127 MessageRS.java	153
Ilustración 128 Modelo físico entidades	175
Ilustración 129 Pantalla login	206
Ilustración 130 Error login	207
Ilustración 131 Panel de registro	207

Ilustración 132 Recuperar contraseña	208
Ilustración 133 Datos de registro	209
Ilustración 134 Estado del servidor de simulaciones	209
Ilustración 135 Estado simulación.....	210
Ilustración 136 Subida casos repetidos	211
Ilustración 137 Vista del repositorio	212
Ilustración 138 Ver solución caso.....	212
Ilustración 139 Re-simular caso	213
Ilustración 140 Descargar caso.....	213
Ilustración 141 Borrar un caso	214
Ilustración 142 Compartir caso (1)	214
Ilustración 143 Compartir caso (2)	215
Ilustración 144 Cerrar sesión.....	215
Ilustración 145 Cambio de idioma.....	216
Ilustración 146 Página siguiente/anterior (1)	216
Ilustración 147 Página siguiente/anterior (2)	217
Ilustración 148 Instalación	220

ÍNDICE DE TABLAS

Tabla 1 Stakeholders	19
Tabla 2 Estudio de la situación actual	21
Tabla 3 Plantilla requisitos de usuario	23
Tabla 4 UR-C00	23
Tabla 5 UR-C01	23
Tabla 6 UR-C02	23
Tabla 7 UR-C03	24
Tabla 8 UR-C04	24
Tabla 9 UR-C05	24
Tabla 10 UR-C06	24
Tabla 11 UR-C07	24
Tabla 12 UR-C08	25
Tabla 13 UR-C09	25
Tabla 14 UR-C10	25
Tabla 15 UR-C11	25
Tabla 16 UR-C12	25
Tabla 17 UR-C13	26
Tabla 18 UR-C14	26
Tabla 19 UR-C15	26
Tabla 20 UR-C16	26
Tabla 21 UR-C17	26
Tabla 22 UR-C18	26
Tabla 23 UR-R00	27
Tabla 24 UR-R01	27
Tabla 25 UR-R02	27
Tabla 26 UR-R03	27
Tabla 27 UR-R04	27
Tabla 28 UR-R05	28
Tabla 29 UR-R06	28
Tabla 30 UR-R07	28
Tabla 31 UR-R08	28
Tabla 32 UR-R09	28
Tabla 33 UR-R10	29
Tabla 34 UR-R11	29
Tabla 35 UR-R12	29
Tabla 36 UR-R13	29
Tabla 37 UR-R14	29
Tabla 38 UR-R15	29
Tabla 39 UR-R16	30
Tabla 40 UR-R17	30
Tabla 41 UR-R18	30

Tabla 42 UR-I00	30
Tabla 43 UR-I01	31
Tabla 44 UR-I02	31
Tabla 45 UR-I03	31
Tabla 46 Estudio alternativa 1.....	35
Tabla 47 Estudio alternativa 2.....	35
Tabla 48 Duración de actividades y tareas.....	43
Tabla 49 Tabla de costes	45
Tabla 50 Tabla ejemplo caso de uso	50
Tabla 51 CU-00	52
Tabla 52 CU-01	53
Tabla 53 CU-02	54
Tabla 54 CU-03	55
Tabla 55 CU-04	56
Tabla 56 CU-05	57
Tabla 57 CU-06	57
Tabla 58 CU-07	58
Tabla 59 Plantilla requisitos software	60
Tabla 60 SR-F00	60
Tabla 61 SR-F01	60
Tabla 62 SR-F02	61
Tabla 63 SR-F03	61
Tabla 64 SR-F04	61
Tabla 65 SR-F05	61
Tabla 66 SR-F06	62
Tabla 67 SR-F07	62
Tabla 68 SR-F08	62
Tabla 69 SR-F09	62
Tabla 70 SR-F11	62
Tabla 71 SR-F11	63
Tabla 72 SR-F12	63
Tabla 73 SR-F13	63
Tabla 74 SR-F14	63
Tabla 75 SR-F15	64
Tabla 76 SR-F16	64
Tabla 77 SR-F17	64
Tabla 78 SR-F18	64
Tabla 79 SR-19	65
Tabla 80 SR-F20	65
Tabla 81 SR-F21	65
Tabla 82 SR-C00.....	65
Tabla 83 SR-C01.....	66
Tabla 84 SR-C02.....	66
Tabla 85 SR-D00	66
Tabla 86 SR-D01	66

Tabla 87 SR-S00	67
Tabla 88 SR-S01	67
Tabla 89 SR-S02	67
Tabla 90 SR-R03.....	67
Tabla 91 SR-M00	68
Tabla 92 SR-M01	68
Tabla 93 SR-M02	68
Tabla 94 SR-M03	68
Tabla 95 SR-M04	69
Tabla 96 SR-M05	69
Tabla 97 SR-M06	69
Tabla 98 SR-R00.....	70
Tabla 99 SR-R01.....	70
Tabla 100 SR-R02.....	70
Tabla 101 SR-R03.....	70
Tabla 102 SR-R04.....	70
Tabla 103 SR-I00.....	71
Tabla 104 SR-I01.....	71
Tabla 105 SR-I02.....	71
Tabla 106 SR-I03.....	72
Tabla 107 SR-O00	72
Tabla 108 SR-O01	72
Tabla 109 SR-O02	72
Tabla 110 SR-O03	73
Tabla 111 SR-O04	73
Tabla 112 SR-C05.....	73
Tabla 113 SR-O06	74
Tabla 114 SR-O07	74
Tabla 115 SR-O08	75
Tabla 116 Matriz trazabilidad UR-SR (1)	76
Tabla 117 Matriz trazabilidad UR-SR (2)	77
Tabla 118 Responsabilidades y atributos tabla ejemplo.....	91
Tabla 119 clase Controller.....	91
Tabla 120 clase Idioma	91
Tabla 121 clase Page	92
Tabla 122 clase Borrar.....	92
Tabla 123 clase Run.....	92
Tabla 124 clase Descargar	92
Tabla 125 clase Ver	92
Tabla 126 clase Upload	93
Tabla 127 clase Registro.....	93
Tabla 128 clase Cuenta.....	93
Tabla 129 clase Logout.....	93
Tabla 130 clase Recuperar	93
Tabla 131 clase Login	94

Tabla 132 clase Activar	94
Tabla 133 clase Share	94
Tabla 134 clase Usuario.....	94
Tabla 135 clase Caso	95
Tabla 136 clase Categoría.....	95
Tabla 137 clase Categoría.....	95
Tabla 138 clase DeleteCaso.....	95
Tabla 139 clase HeartBeat.....	95
Tabla 140 clase Simulación	96
Tabla 141 Asociaciones y agregaciones tabla ejemplo	96
Tabla 142 Asociación Usuario-Caso	96
Tabla 143 Asociación Usuario-Share	96
Tabla 144 Asociación Caso-Share	96
Tabla 145 Asociación Usuario-DeleteCaso.....	97
Tabla 146 Caso-Categoría.....	97
Tabla 147 Generalización Caso	97
Tabla 148 Generalización DeleteCaso	97
Tabla 149 Perfiles y diálogos.....	100
Tabla 150 Símbolos mapa de navegación	107
Tabla 151 Matriz de trazabilidad CU-UI	109
Tabla 152 Excepción plantilla	112
Tabla 153 EX-00.....	112
Tabla 154 EX.01	113
Tabla 155 EX-02	113
Tabla 156 EX-03.....	113
Tabla 157 EX-04.....	113
Tabla 158 EX-05.....	114
Tabla 159 Componente vista.....	118
Tabla 160 Componente controlador	118
Tabla 161 Componente modelo.....	118
Tabla 162 Clases asociadas a casos de uso	120
Tabla 163 UI-00	131
Tabla 164 UI-01	131
Tabla 165 UI-02	134
Tabla 166 UI-03	136
Tabla 167 UI-04	137
Tabla 168 UI-05	139
Tabla 169 UI-06	139
Tabla 170 UI-07	140
Tabla 171 Revisión CU-Subsistemas-UI	141
Tabla 172 atributos SeeServlet.java.....	154
Tabla 173 atributos UploadServlet.java	154
Tabla 174 atributos User.java	155
Tabla 175 atributos Project.java	155
Tabla 176 atributos Sharing.java.....	156

Tabla 177 atributos Deletedproject.java.....	156
Tabla 178 atributos Statuscategory.java.....	156
Tabla 179 atributos ProxyManager.java	156
Tabla 180 atributos Funciones.java	157
Tabla 181 atributos Comunicacion.java	157
Tabla 182 atributos Simulate.java.....	157
Tabla 183 atributos MessageRQ.java.....	158
Tabla 184 atributos MessageRS.java.....	158
Tabla 185 métodos ServletController.java.....	159
Tabla 186 métodos AccountServlet.java.....	159
Tabla 187 métodos ActivateServlet.java	159
Tabla 188 métodos DeletedSetvlet.java	160
Tabla 189 métodos DownloadServlet.java	160
Tabla 190 métodos LanguageServlet.java	160
Tabla 191 métodos LoginServlet.java	161
Tabla 192 métodos LogoutServlet.java.....	161
Tabla 193 métodos RunServlet.java.....	162
Tabla 194 métodos RegisterServlet.java.....	162
Tabla 195 métodos PageServlet.java	162
Tabla 196 métodos RecoverServlet.java	163
Tabla 197 métodos SeeServlet.java	163
Tabla 198 métodos ShareServlet.java.....	164
Tabla 199 UploadServlet.java.....	164
Tabla 200 métodos User.java.....	165
Tabla 201 métodos Project.java.....	165
Tabla 202 métodos Sharing.java	166
Tabla 203 métodos Deletedproject.java	166
Tabla 204 métodos Statuscategory.java	166
Tabla 205 métodos ProxyManager.java.....	167
Tabla 206 métodos Funciones.java	170
Tabla 207 métodos Comunicacion.java	171
Tabla 208 métodos HeartBeat.java.....	171
Tabla 209 Simulate.java	172
Tabla 210 métodos MessageRQ.java	173
Tabla 211 métodos MessageRS.java	173
Tabla 212 Tablas BD	175
Tabla 213 Tabla Users	176
Tabla 214 Tabla Projects	176
Tabla 215 Tabla Deletedprojects.....	176
Tabla 216 Tabla Statuscategories.....	176
Tabla 217 Tabla Sharings.....	177
Tabla 218 Matriz Clases diseño/Casos de uso reales.....	179
Tabla 219 Matriz arquitectura del sistema/Entorno tecnológico	179
Tabla 220 Matriz clases/datos	180
Tabla 221 Matriz Vista/Requisitos Software	181

Tabla 222 Matriz Controlador/Requisitos Software	182
Tabla 223 Matriz Modelo/Requisitos Software	183
Tabla 224 PU-00	186
Tabla 225 PU-01	186
Tabla 226 PU-02	186
Tabla 227 PU-03	186
Tabla 228 PU-04	187
Tabla 229 PU-05	187
Tabla 230 PU-06	187
Tabla 231 PU-07	187
Tabla 232 PU-08	187
Tabla 233 PU-09	188
Tabla 234 PU-10	188
Tabla 235 PU-11	188
Tabla 236 PU-12	188
Tabla 237 PU-13	189
Tabla 238 PU-14	189
Tabla 239 PU-15	189
Tabla 240 PU-16	189
Tabla 241 PU-20	190
Tabla 242 PA-00	190
Tabla 243 PA-01	190
Tabla 244 PA-02	191
Tabla 245 PA-03	191
Tabla 246 PA-04	191
Tabla 247 PA-05	191
Tabla 248 PA-06	192
Tabla 249 PA-07	192
Tabla 250 PA-08	192
Tabla 251 PA-09	192
Tabla 252 PA-10	193
Tabla 253 PS-00	193
Tabla 254 PS-01	193
Tabla 255 PS-02	194
Tabla 256 PI-00	194
Tabla 257 PI-01	195
Tabla 258 PI-02	195
Tabla 259 PN-00	196
Tabla 260 PN-01	196
Tabla 261 PN-02	196
Tabla 262 PN-03	197
Tabla 263 PN-04	197
Tabla 264 PN-05	197
Tabla 265 PN-06	198
Tabla 266 PN-07	198

Tabla 267 PN-08	199
Tabla 268 PN-09	199
Tabla 269 PN-10	199
Tabla 270 PN-11	200
Tabla 271 PN-12	200
Tabla 272 PN-13	200
Tabla 273 PN-14	201
Tabla 274 PU/SR	201
Tabla 275 PA/UR	202
Tabla 276 PS/SR.....	202
Tabla 277 PI/SR	202
Tabla 278 PN/SR.....	202

1. Abstract

Nowadays, all major businesses and institutions use the combination of computing and Internet as a common working tool. The main objective of this combination is improving efficiency and effectiveness in the daily work of its employees. From the perspective of the organization, it implies a highly reduction in stored physical documents and a great opportunity of increasing the turnover.

At the present time, there are few companies that do not use cloud storage, ERP or own intranet in order to improve company's performance and management. These companies are restricting their options of growing in the business world and are in danger of disappearing in the near future because of its poor technological level.

Web applications are an important part of this topic. Web applications are a big alternative to the client software that is necessary install in each of the employee's terminals of a company. Therefore, this way it is possible to save considerable storage space in computers and is more comfortable and fast for employees because they only need a browser and a URL to start to use the application.

The bachelor thesis contained in this document explains the development of a web application in order to use simulation tools. The author of this thesis is Javier García Pérez, Bachelor's Degree in Computer Science and Engineering of Universidad Carlos III, and the director is Israel González Carrasco. This bachelor thesis follows the development methodology Métrica 3[5] with the goal of doing this document as complete, structured and readable as possible. This document contains the main stages of this methodology: system requirements, project estimation and management, system analysis, system design, etc. Furthermore, the web application code will be implemented in the programming language Java with Eclipse as IDE.

The summary of the main features of this application is listed below. These features will be performed by the functionality of the classes that are explained in detail in this document:

- **User Management**
 - *User signup*: any new user who will use the application must signup completing a web form.
 - *User login*: any user must login with his email and password to access to the web application.
 - *User profile*: any user can modify his information signup after login, except his email.
- **Project Management**
 - *Project upload*: any user can upload simulation projects after login. These simulation projects shall meet some standards. Projects will be stored on the server and users can access them from his repository.
 - *Project simulation*: users can start a simulation of an uploaded project.

- *Project sharing*: users can sharing a project, after finishing its simulation, with other registered users in the web application. The owner of the sharing can manage permissions on the project of the shared user.
- *Project download*: users can download a project after finishing its simulation. The download will contain the solution of the project.
- *Project deleted*: Users can delete a project after finishing its simulation or if it has errors. The project will be deleted from the user repository but it will be moved to a deleted projects folder on the server.
- *Project solution*: users can see the project solution from the application without needing to download the project previously.

The simulation process will be performed by different servers. These servers contain simulation algorithms to resolve submitted projects.

Our web application communicates with the servers through Web Services. This service has all necessary for completing communication between servers during simulation process. Furthermore, Web Services of the web application also do a heartbeat process, which will be responsible for knowing the status of simulation servers.

Nowadays, being able to access your documents from anywhere and device is very important and this application allow this because your simulation projects are on a server and not in a local terminal. Furthermore, the tedious task of having to share a simulation project with each member of the teamwork disappears due to the application itself can do this without leaving the application. In this way, all users with access to the project always see the most current project version. Notice that use any of the features listed above does not block other functions of the application. Therefore, users can perform other tasks, for example, at the time a project is being simulated.

The objectives of this project are explained below. This system came from the need to manage simulations projects in a comfortable and simple way, allowing to launch the simulation and observe the results from the web application, without using outside programs to perform some of these functions.

The project is not only a file repository, as can be understood initially, but is a set of features that allow users and teamworks to observe, simulate, share, download, upload and delete simulation projects from a single web application.

A main objective of this project is test an initial version of the web application and in this way assess problems that may arise as computing workload, bandwidth, concurrency or IT security. This will allow assessing to their feasibility and deciding whether in the future it is possible to expand the system with functionality that is more complex.

Another objective is to see the performance of the application on its Web Services communications with simulation servers. The project will provide users a GUI to simulate their uploaded projects on external servers. These servers will contain the algorithms for calculating the solution. Synchronization between the web application and simulation servers will be analysed, verify that the application knows at any time whether simulation servers are

operational and see whether the user when launches a simulation can check in real time this progress from his interface.

A goal considered at the time of developing the application has been get an optimal iteration between the user and the GUI. It has been focused on an easy front-end design as simple as possible to users that use the application does not feel lost at any time and know that steps are being followed.

To develop the system and write this document have been used the following tools and software:

- Eclipse Luna IDE for Java EE Developers
- Microsoft Office Professional Plus 2013
- Microsoft Windows 8.1
- Java SE Development Kit 7
- WildFly 8.2.0 Final
- MySQL Enterprise Edition 5.6.24.0
- Google Chrome
- Notepad++

The development of this project began from studying customer needs. From these needs have been written a number of requirements that must satisfy both parties (client and Project Manager). These requirements are the foundation of the project developed.

User requirements written in this document are of three types. The features of each are described below:

- **Capacity requirements:** They represent what users need to solve a problem or achieve a goal.
- **Restriction requirements:** They are the restrictions imposed by users about how to solve the problem or how to achieve the goal.
- **Reverse requirements:** They contain all points that the web application should not do under any circumstances.

Considering these requirements the two alternative solutions assessed for the project are Java Servlets and PHP, both with MySQL as database. Java Servlets technology allows a simple way to control all incoming requests to the server using Servlets or filters. It can even be extend the functionality of Servlets adding code in EJB (Enterprise Java Beans). All data that the servlet has processed are then transferred to JSP pages that show the user the information in a dynamic way. The PHP code is executed on the server generating the resultant HTML that is sent to the client. One of the most attractive things of this language is its simplicity for people with little knowledge in this area because it is a weakly-typed language.

Regarding the choice of Java Servlets instead of PHP for the project architecture it is believed to have been the right choice for what was intended and best suited to the chosen meta-pattern MVC + Front Controller. Moreover, in view of future improvements and extensions, Java Servlets is a more scalable alternative that PHP.

MVC (Model-View-Controller) is a software architectural pattern for implementing user interfaces. It divides a given software application into three interconnected parts, so as to separate internal representations of information from the ways that information is presented to or accepted from the user. This software architectural pattern defines the independent organization model (business objects), view (GUI) and the controller (workflow application). Characteristics of the three parties in summary are:

- Model
 - It contains the core application functionality.
 - Encapsulates the state of the application.
 - Independent of the controller and the view.
- View
 - It is the presentation of the model.
 - It can access the model but never change its state.
 - It can be notified when there is a change of state in the model.
- Controller
 - Reacts to the client request by executing the appropriate action and creating the appropriate model.

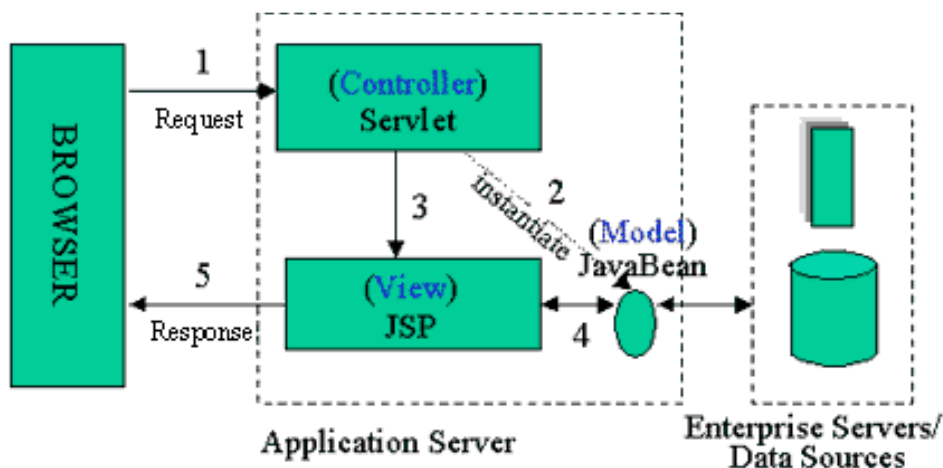


Illustration 1 MVC - Java Servlets | Source: <http://www.javaworld.com/article/2076557/java-web-development/understanding-javascript-model-2-architecture.html>

Front-Controller pattern[9] recommends the use of a centralized point of the system for the treatment of access requests. This is because if it does not have a centralized access the navigation of system is defined in the view or JSP pages, which produces duplicate code and difficult to maintain because to the mixture of presentation with navigation.

This pattern in the environment of a web application involves having a controller servlet as a single point of access to the application, this does not produce a system output, only redirects the request to a sub controller. This centralizes all incoming client requests, forwarding every request to the appropriate manager or dispatcher that will manage client response.

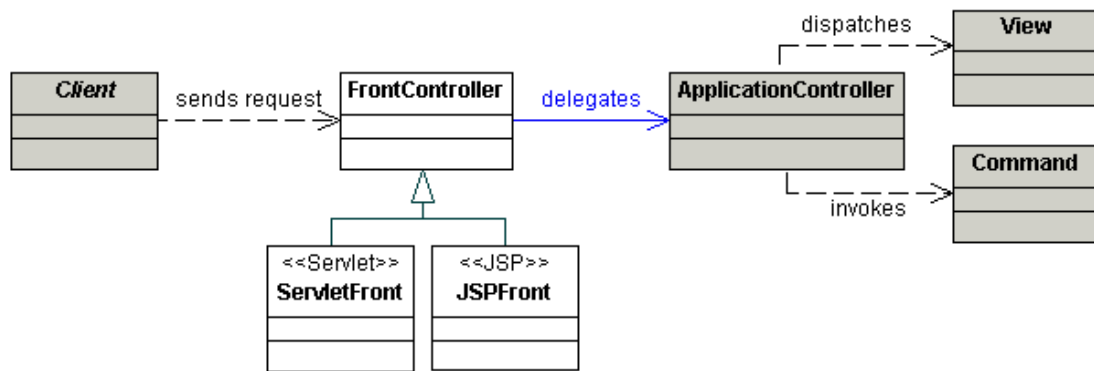


Illustration 2 Front-Controller | Source: <http://corej2eepatterns.com/FrontController.htm>

In our web application, the union of these two patterns would result in the following structure.

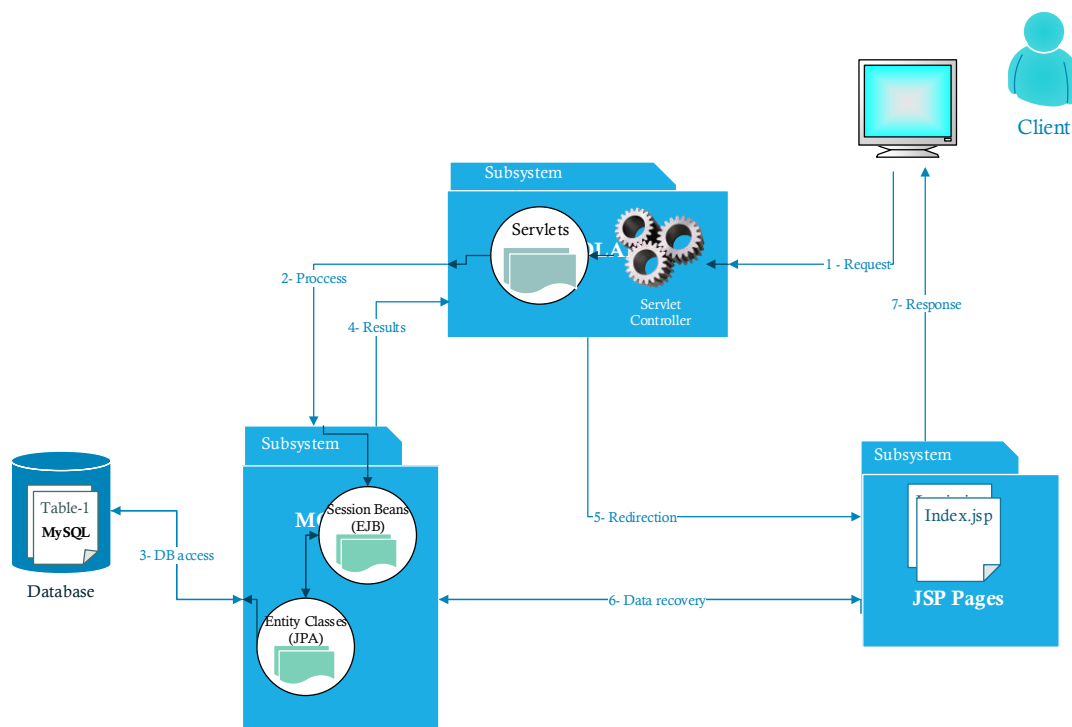


Illustration 3 MCV + Front Controller

In short, to our system, any request will arrive at controller and will be managed by the servlet-controller, which will redirect the request to the appropriate servlet. It will process the request by sending it to its EJB, which, aided by JPA to interact with database, will achieve results that will be returned to the controller and it will send to the view where the user can see the result of his request.

Clarify that this project is a first version of the system and does not share modules with other subsystems, so the case of sharing and reuse of EJBs is not given. It was decided that the business logic is implemented within servlets to which redirect the servlet-controller, in order to reduce complexity. JPA will still be used to communicate with the database.

Project planning has been estimated based on the life cycle chosen. After obtaining the number of days, the tasks and activities of the project have been adjusted within this range. All this information will be displayed fully in a diagram GANTT. This diagram shows the group of activities divided into their respective tasks. Each task will have a start date and end date that must be fulfilled during development.

The software project is composed of the following activities that it listed and summarized below:

- **Project management:** Planning and estimation of project activities and tasks are performed. In addition, the project life cycle is chosen to continue to perform the following stages.
- **Feasibility study:** User requirements are identified based on the information gathered from the client. Based on the requirements we study, analyse and select an alternative for the solution of the system. Subsequently, with this information, it is determined whether the system is feasible.
- **System analysis:** The software requirements are identified, it takes place to high standards the architecture definition to follow. Finally, different use cases are identified based on the requirements.
- **System design:** It builds on the analysis of the system. In the system design the GUI are defined, the model of the database to use is studied. Finally, the required specifications are generated for the coding of the web application.
- **System development:** It focuses on the coding of all aspects included in the system analysis and later expanded into the system design. Finally is completed with the drafting of the user manual.
- **System integration:** This activity consists in verifying the correct and complete operation of the system by using test plan.
- **Deadline:** Agreed deadline to send the system to the client

Regarding starting dates, finish dates and days duration of the activities you can see in the following table:

Activities	Duration	Starting	Ending
Project management	110 days	Mon 01/12/15	Fry 06/12/15
<i>Life cycle</i>	<i>11 days</i>	<i>Mon 01/12/15</i>	<i>Mon 01/26/15</i>
<i>Planning</i>	<i>99 days</i>	<i>Tue 01/27/15</i>	<i>Fry 06/12/15</i>
<i>Estimation</i>	<i>11 days</i>	<i>Mon 02/02/15</i>	<i>Mon 02/16/15</i>
<i>Configuration management</i>	<i>110 days</i>	<i>Mon 01/12/15</i>	<i>Fry 06/12/15</i>
<i>Quality management</i>	<i>110 days</i>	<i>Mon 01/12/15</i>	<i>Fry 06/12/15</i>
Feasibility study	12 days	Mon 01/12/15	Tue 01/27/15
<i>Determine the extent</i>	<i>2 days</i>	<i>Mon 01/12/15</i>	<i>Tue 01/13/15</i>
<i>User requirements</i>	<i>3 days</i>	<i>Wed 01/14/15</i>	<i>Fry 01/16/15</i>
<i>Study and choice of the alternative</i>	<i>7 days</i>	<i>Mon 01/19/15</i>	<i>Tue 01/27/15</i>
System analysis	19 days	Wed 01/28/15	Mon 02/23/15
<i>Use case</i>	<i>8 days</i>	<i>Wed 01/28/15</i>	<i>Fry 02/06/15</i>
<i>Software requirements</i>	<i>11 days</i>	<i>Wed 01/28/15</i>	<i>Wed 02/11/15</i>
<i>Architecture analysis</i>	<i>8 days</i>	<i>Thu 02/12/15</i>	<i>Mon 02/23/15</i>
System design	30 days	Tue 02/24/15	Mon 04/06/15

<i>GUI desing</i>	<i>13 days</i>	<i>Tue 02/24/15</i>	<i>Thu 03/12/15</i>
<i>Study of the DB</i>	<i>11 days</i>	<i>Mon 03/02/15</i>	<i>Mon 03/16/15</i>
<i>Define the architecture</i>	<i>20 days</i>	<i>Tue 03/10/15</i>	<i>Mon 04/06/15</i>
System development	39 days	Tue 04/07/15	Fry 05/29/15
<i>GUI coding</i>	<i>11 days</i>	<i>Tue 04/07/15</i>	<i>Tue 04/21/15</i>
<i>Create DB</i>	<i>8 days</i>	<i>Tue 04/07/15</i>	<i>Thu 04/16/15</i>
<i>Coding the functionality</i>	<i>23 days</i>	<i>Wed 02/22/15</i>	<i>Fry 05/22/15</i>
<i>User manual</i>	<i>5 days</i>	<i>Mon 05/25/15</i>	<i>Fry 05/29/15</i>
System integration	9 days	Mon 06/01/15	Thu 06/11/15
<i>Unit test</i>	<i>9 days</i>	<i>Mon 06/01/15</i>	<i>Thu 06/11/15</i>
<i>Integration test</i>	<i>9 days</i>	<i>Mon 06/01/15</i>	<i>Thu 06/11/15</i>
Deadline	0 days	Fry 12/06/15	Fry 06/12/15

Table 1 Activities

After the planning, it has been carried out the estimate project costs. The cost estimate has considered only one employee (Project Manager) and salary 16 € / h, with a working day of eight hours

Finally, the estimation made for this project has been adequate and it has been possible comply with the beginnings and endings of the tasks that make up it. Moreover, most importantly, it has respected the deadline.

Before starting the design stage has conducted a system analysis. This analysis shows the standards and regulations to meet, software requirements, use cases, classes, GUI, subsystems, etc. All this information will form the basis for the design where this will be completed to get a final documentation that the programmer can understand well for doing the coding.

Software requirements establish how the system works so that this is correct. There are different kinds, those that we will use in our project are:

- **Functional requirements:** Its aim is to define the purpose of the software and are derived from the use cases, which in turn derived from the capacity user requirements.
- **Quality requirements:** Specify the attributes of the software to ensure that it will be fit for purpose.
- **Documentation requirements:** Set out the specific project requirements for documentation, in addition to those contained in the standards.
- **Safety requirements:** Specify the requirements to secure the system against threats to confidentiality, integrity and availability.
- **Check requirements:** These requirements specify limitations that affect how the software should verify data input and output.
- **Performance requirements:** Assign numerical values for performance variables, such as data transfer rates, frequency, capacity and processing speed.
- **Interface requirements:** Specifies hardware and/or software that the system or system components must interact or communicate
- **Operation requirements:** Are those requirements that will indicate how the system will perform the tasks for which it was built, ensuring required service levels.

In general terms, the purpose of system analysis is a detailed specification of the system to be developed. This stage of analysis is to define the problem to be solved. It is about

understanding the needs to be solved and model the problem using different techniques (depending on the characteristics of this project) to subsequently resolve it in system design.

In the system design, the main objective is to solve the problem that is described and modelled in the stage of system analysis. With this solution, it seeks to define the system architecture and technological environment.

The design contains, among other things, specifications necessary to build the system, the technical description of the test plan and the definition of requirements of implantation of the system.

The design of the database obtained in this stage contains the following tables:

- **Users:** Stores information of registered users in the web application.
 - *UID:* user ID
 - *FirstName:* user name
 - *FamilyName:* user lastname
 - *Email:* user email
 - *IsValidate:* verified account
 - *DateRegistration:* user registration date
 - *Password:* user password
- **Projects:** Stores information of simulation projects uploaded to the application server.
 - *IdProject:* project ID
 - *ProjectName:* project name
 - *ProjectDescription:* project description
 - *ONGFile:* project path in the simulation server
 - *UID:* ID of the user owner
 - *IdProjectStatus:* simulation status
 - *DateCreation:* upload date
 - *DateModifeied:* last simulation date
- **Deletedprojects:** Stores information of simulation projects deleted in the web application.
 - *IdProject:* project ID
 - *ProjectName:* project name
 - *ProjectDescription:* project description
 - *ONGFile:* project path in the simulation server
 - *UID:* ID of the user owner
 - *IdProjectStatus:* simulation status
 - *DateCreation:* upload date
 - *DateDeleted:* delete date
- **Statuscategories:** Stores information on the types of status that can take a simulation project.
 - *IdProjectStatus:* status ID
 - *StatusName:* status name
 - *StatusDescription:* status description
- **Sharings:** Stores information of sharing made by users on simulation projects.
 - *IdSharing:* sharing ID

- *IdProject*: project ID
- *UID*: shared user ID
- *UIDsharer*: user ID who made the sharing
- *DateShared*: sharing date that was created
- *DateChanged*: last sharing date modified
- *AllowRecalculate*: permission to recalculate
- *AllowDelete*: permission to delete
- *AllowDownload*: permission to download
- *AllowShare*: permission to share

The relation between the above tables are shown in the following diagram UML:

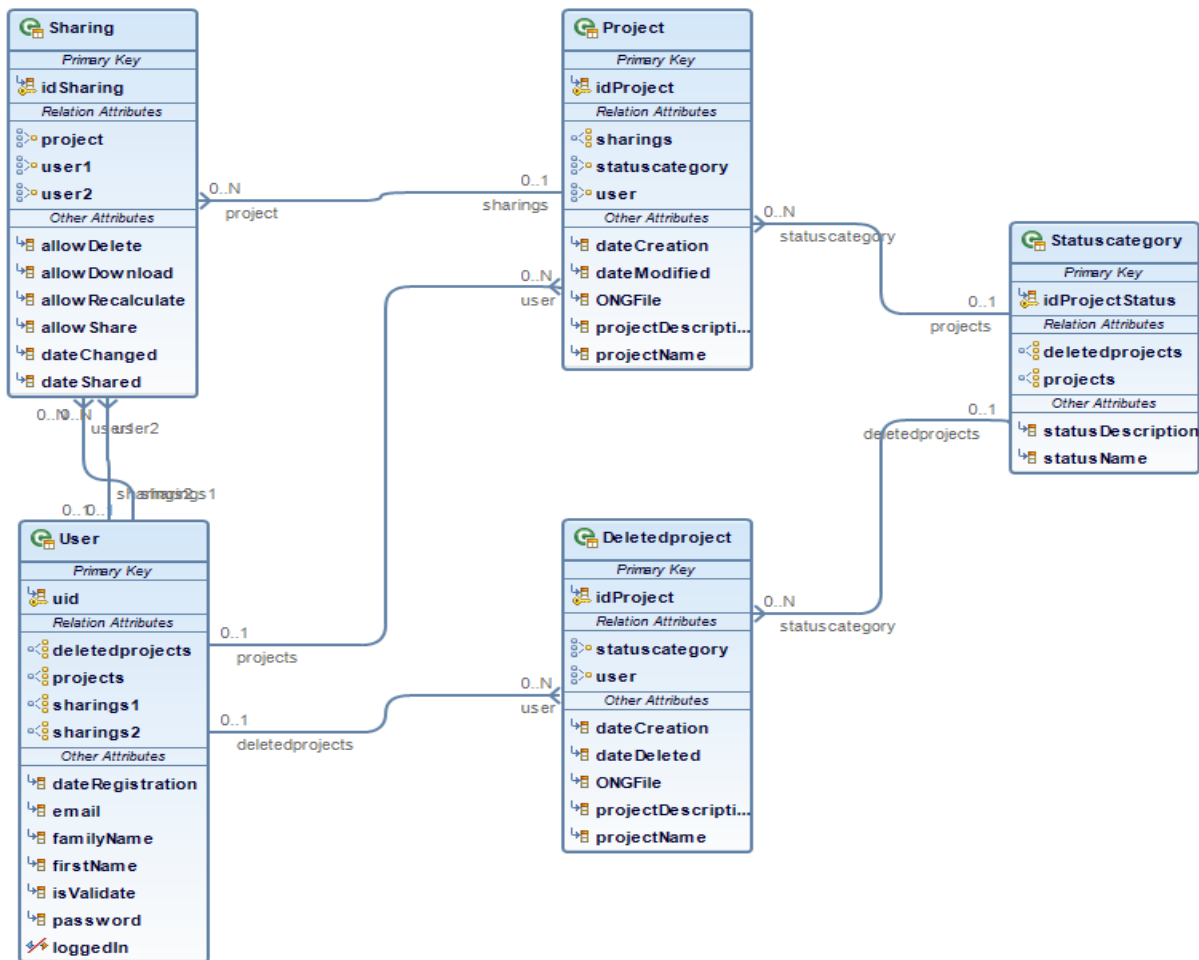


Illustration 4 database diagram UML

The fully design contained in this document has helped satisfactorily to the subsequent codification because has accelerated fairly this process when having the foundations of what we wanted to do.

The main objective of the test plan is to be complete and allow checking that the system works properly and meets the specifications set along document.

This task will verify and validate the system for the client. It checks whether the requirements and specifications are met, without defects and so on.

All requirements have been met and verified in the various matrices and consistency analysis that have been done in the various paragraphs of this document.

Overall, the project was complete and satisfactory in terms of specified requirements and the given time.

Looking ahead, there are some improvements that could be implemented in the web application. Although the first version of the application is quite complete, it should never stop thinking about adding new extensions that can improve or even complete more functionality. The improvements considered for this system are:

- **Adding more languages:** The application already supports English and Spanish languages, but would be more complete by adding other popular languages to reach more people. Keep in mind that there may be people interested in the web application that does not understand English or Spanish so adding a language who they understand would facilitate the use of the application.
- **Improving accessibility:** Although this point has been respected in the choice of colours, font size, icons sizes, etc. it can be improved further by adding other features to make the application accessible to other people. These improvements are the correct structure of the HTML code to make it compatible with screen readers, avoid that the elimination of CSS style leaves the page inaccessible, etc.
- **Responsive design:** Maybe one of the most important. The application would have more use if its design allowed using it efficiently from different electronic devices (smartphones, tablets, etc.) with different resolutions. Nowadays, many people use their smartphone as a working tool, so that this improvement would open up the doors to a significant number of new users.
- **Support for more types of simulation projects:** This first version uses a specific type of simulation projects that comply the testing set. This point can be improved by introducing application compatibility with other simulation projects that could be simulated in their respective simulation servers. For reach this goal, it would be necessary to add the small functionality to establish Web Services between simulation servers and web application. Remember that the system currently uses a simulation test server.
- **Setting maximum storage space per user:** For this first version of the system it has not been taken into consideration the total storage occupied by simulation projects uploaded to the server for each user. In future releases it would be suitable for putting a limit in order to avoid that the performance of the application get worse or make impossible that malicious users can collapse the storage capacity of the server.

Remember that these improvements were not in the requirements and that the web application in this version works correctly as specified and included in the document.

2. Introducción

A día de hoy toda empresa e institución que se precie combina la informática e internet como una herramienta más de trabajo, cuyo principal objetivo es mejorar la eficacia y eficiencia en las labores de sus trabajadores y, a nivel de organización, una reducción notable de la documentación física almacenada y una gran oportunidad de aumentar su volumen de negocio. En la actualidad hay pocas empresas que no almacenen información en la nube, dispongan en su intranet de aplicaciones específicas o posean de un ERP para la autogestión de la propia empresa.

Adentrándonos en las aplicaciones web, estas son una gran alternativa a los programas clientes que son necesarios tener instalados en cada uno de los terminales de trabajo que usan los empleados. De esta manera se ahorra en espacio dentro de las computadoras y se proporciona un acceso más cómodo para el empleado, ya que únicamente tiene que usar su navegador web y una dirección como llave para empezar a interactuar con la aplicación.

El proyecto fin de grado que recoge este documento trata sobre una aplicación web que proporciona a cada usuario un repositorio en el que podrá gestionar casos de simulación, ejecutarlos en un servidor ajeno, que cuenta con la funcionalidad para realizar esta tarea, y tras finalizar la operación poder gestionar el resultado. Además, los usuarios podrán descargar la solución para guardarlo en otros equipos. Todo esto se podrá realizar desde una única aplicación web que sólo pedirá un registro y login previo para poder usar todas sus funcionalidades.

Puntualizando más en los detalles la aplicación proporciona:

- Creación y gestión de cuentas de usuario: edición de datos de registro, recuperación de contraseña,...
- Repositorio de almacenamiento de casos de simulación en el servidor de la aplicación.
- Lanzar simulación de los casos contra el servidor ajeno.
- Gestión de proyectos propios, permitiendo la descarga de estos, la posibilidad de compartir con otros usuarios registrados, ver el resultado, relanzar simulación, borrar...

En la actualidad, el poder acceder a tus documentos desde cualquier lugar y dispositivo es clave, y esta aplicación web ayuda en esta tarea ya que permite acceder a los casos de simulación sin depender del dispositivo usado al estar estos alojados en el servidor. Por otro lado, elimina la pesada tarea de tener que mandar los proyectos a todos los miembros del grupo de trabajo debido a que se permite la compartición de estos desde la propia aplicación web, y así, de esta manera, todos los usuarios que tengan acceso al caso siempre podrán ver su versión más actual. Por otro lado, puntualizar que los usuarios que lancen la simulación de un caso, como los que tengan acceso a él, no sufrirán ningún bloqueo de las funcionalidades de la aplicación hasta que esta tarea finalice, sino que podrán realizar otros menesteres dentro de la aplicación.

Para finalizar esta introducción aclarar que la estructura de este documento seguirá la metodología de desarrollo Métrica 3[5] con el objetivo de hacerlo accesible y comprensible a los lectores.

2.1. Objetivos

Este proyecto surge de la necesidad de poder gestionar casos de simulación de una manera cómoda y sencilla. Permitir lanzar su simulación y ver sus resultados desde una misma aplicación web sin la necesidad de tener que usar programas ajenos para realizar cada una de las funciones descritas anteriormente.

El proyecto no busca ser tratado como un simple repositorio online de casos, sino como un comprendido de funcionalidades que permiten englobar en una única aplicación las posibilidades de simular, compartir, descargar, ver resultados, ... de los casos de simulación de una persona o de un equipo de trabajo.

Uno de los principales objetivos del proyecto es evaluar en su versión inicial problemas que puedan darse de carga computacional, anchos de banda, concurrencia o seguridad informática y de esta manera poder valorar su viabilidad y decidir si en un futuro es posible ampliar el sistema con funcionalidades mucho más complejas.

Otro de los principales objetivos es ver en profundidad cómo se comporta la aplicación en sus comunicaciones Web Services con el servidor de simulaciones, se recuerda que el proyecto servirá de puente al usuario para simular sus casos subidos contra un servidor ajeno el cual tendrá los algoritmos para el cálculo de la solución. Se tendrá que analizar la sincronización que existe entre la aplicación web y el servidor de simulación, comprobar que la aplicación sabe en todo momento si el servidor de simulación está operativo y ver si el usuario a la hora de lanzar una simulación puede comprobar en tiempo real el progreso de esta desde su interfaz.

Una de las principales características que se ha buscado a la hora de desarrollar esta aplicación ha sido conseguir una óptima iteración entre el usuario y la GUI. Para ello se ha realizado un diseño sencillo del front-end con la mayor simplicidad posible para lograr que los usuarios a la hora de usar la aplicación no se pierdan y sepan en todo momento que pasos están siguiendo.

2.2. Principales herramientas utilizadas

A continuación se detallan los principales programas y herramientas que han sido utilizados para el desarrollo del TFG:

- **Eclipse Luna IDE for Java EE Developers:** Herramienta de desarrollo usada para la elaboración del código de la aplicación.
- **Microsoft Office Professional Plus 2013:** Conjunto de programas que han servido para realizar toda la documentación relacionada con el proyecto (memoria, diagramas, esquemas, planificación,...)
- **Microsoft Windows 8.1:** Sistema operativo usado durante la elaboración del proyecto.
- **Java SE Development Kit 7:** Conjunto de herramientas y componentes que permiten el desarrollo en java de la aplicación.
- **WildFly 8.2.0.Final:** Servidor de aplicaciones donde se desplegará la aplicación.

- **MySQL Enterprise Edition 5.6.24.0:** Conjunto de herramientas para la gestión de la base de datos.
- **Google Chrome:** Navegador web principal donde se probará la aplicación
- **Notepad++:** Editor de textos usado para cambiar fragmentos simples de código.

3. Definiciones y Acrónimos

Esta sección contiene el significado de ciertas definiciones y acrónimos que se encontrarán a lo largo de la lectura de este documento. El objetivo es aclarar su significado para evitar confusiones al lector.

3.1. Definiciones y acrónimos

A continuación se muestran las definiciones y acrónimos que se irán encontrando a lo largo del documento:

- **Aplicación Web (app):** cualquier software que se ejecuta desde un navegador web. Es creado en un lenguaje de programación soportado por los navegadores (ej. HTML+CSS+JS).
- **CSS:** lenguaje usado para definir y crear la presentación de un documento estructurado escrito en HTML o XML.
- **Drag&Drop:** expresión usada en el mundo de la informática para expresar la acción de arrastrar con el ratón un objeto de una ventana a otra o entre partes de una misma ventana.
- **EVS:** estudio de viabilidad del sistema.
- **Front-Controller pattern:** patrón de diseño software que se basa en la centralización en un punto de la aplicación de la recepción de las solicitudes (*request*).
- **Front-End:** parte de la aplicación software que interactúa con los usuarios.
- **GANTT:** diagrama cuyo objetivo es exponer el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total determinado durante un proyecto.
- **GET:** uno de los muchos métodos de request soportados por el protocolo HTTP utilizado por la World Wide Web.
- **Heartbeat:** servicio web cuya función es comprobar si el destinatario de la request está operativo.
- **HTML:** lenguaje de marcado para la elaboración de páginas web.
- **HTTP:** es el protocolo usado en cada transacción de la World Wide Web
- **JAVA EE:** plataforma de programación para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java, anteriormente conocida como J2EE.
- **Java Servlet:** clase en el lenguaje de programación Java, utilizada para ampliar las capacidades de un servidor.
- **JavaScript (JS):** lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.
- **jQuery:** biblioteca de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.
- **JSP:** JavaServer Pages es una tecnología que ayuda a los desarrolladores de software a crear páginas web dinámicas basadas en HTML, XML, entre otros tipos de documentos.

- **MySQL:** sistema de gestión de bases de datos relacional, multihilo y multiusuario.
- **ONGF:** extensión de los proyectos que se subirían a la aplicación.
- **POST:** uno de los muchos métodos de request soportados por el protocolo HTTP utilizado por la World Wide Web.
- **Response:** después de recibir e interpretar un mensaje request un servidor responde con un mensaje response.
- **Request:** mensaje de solicitud enviado por un cliente a un servidor.
- **TFG:** Trabajo Fin de Grado.
- **Web Services:** método de comunicación entre dos dispositivos electrónicos a través de la red.
- **WildFly:** servidor de aplicaciones que implementa la especificación Java EE.
- **XML:** lenguaje de marcas desarrollado por el World Wide Web Consortium.
- **XSD:** lenguaje de esquema utilizado para describir la estructura y las restricciones de los contenidos de los documentos XML de una forma muy precisa

4. Estudio de viabilidad del sistema (EVS)

Este apartado estudia el alcance de la necesidad planteada por el cliente realizando una descripción general teniendo presente las restricciones económicas, técnicas, legales y operativas. Se analizarán los objetivos, el alcance y los grupos interesados en el sistema (*stakeholders*).

En este punto se describirá el sistema y las alternativas necesarias de cara al desarrollo del proyecto, así, como la solución final elegida y el porqué de ella.

4.1. Establecimiento del alcance del sistema

Recoge el estudio del alcance de las necesidades planteadas por el cliente. A partir de las necesidades vistas se identificarán unos requisitos que deben satisfacer a ambas partes y se identifican las unidades organizativas afectadas estableciendo su estructura.

4.1.1. Estudio de la solicitud

El cliente solicitó el desarrollo de una aplicación para el uso de una herramienta de simulación, la cual permita la gestión de los casos almacenados.

La aplicación será accesible a través desde un navegador web y usará el protocolo de transacción http.

La aplicación contará principalmente con dos módulos:

- Módulo de usuarios:
 - Login y registro de usuarios.
 - Gestión de datos del usuario.
 - Sistema de activación y recuperación de cuenta.
- Módulo de casos:
 - Almacenamiento de casos en el servidor.
 - Gestión de los casos subidos.
 - Simulación de casos subidos.
 - Consulta de resultados de simulación.
 - Descarga de casos simulados.
 - Compartición de casos simulados.

En definitiva, el cliente propone una aplicación web con unas características que hacen que la realización del proyecto sea viable

Este trabajo será supervisado y guiado por Israel González Carrasco y realizado por Javier García Pérez, estudiante de la titulación de Ingeniería Informática en la Universidad Carlos III de Madrid.

4.1.2. Identificación del alcance del sistema

Los objetivos y principales funcionalidades del comportamiento del sistema se recogen en el siguiente diagrama de flujo y se explican en detalle después de este:

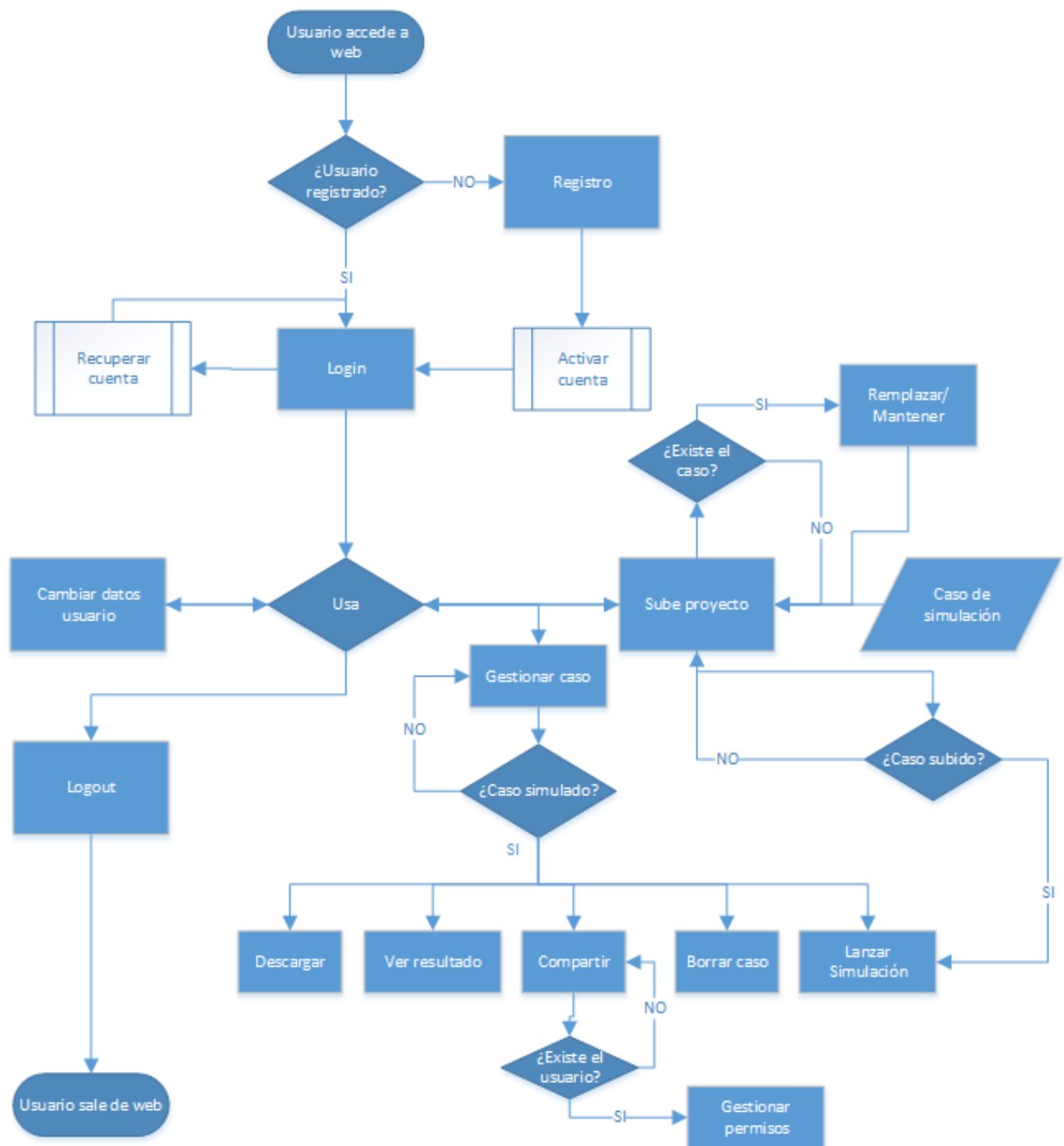


Ilustración 5 Diagrama de flujo de la App

El diagrama anterior muestra los principales pasos y usos que un usuario debe seguir para realizar una determinada función en la aplicación web.

- **Registro:** El usuario para acceder a la aplicación deberá registrarse previamente completando un formulario.
 - *Activar cuenta:* Tras el registro el usuario deberá activar su cuenta siguiendo los pasos del correo recibido[8] en su email.
- **Login:** Si el usuario ya dispone de una cuenta deberá introducir el email y contraseña de esta para acceder.
 - *Recuperar cuenta:* Si el usuario tiene una cuenta pero no recuerda la contraseña podrá recuperarla siguiendo los pasos del correo recibido en su email[8].
- **Cambiar datos de usuario:** Una vez logueado el usuario podrá cambiar sus datos de registro (a excepción del email).
- **Subir proyecto:** El usuario podrá subir casos de simulación que se almacenarán en el servidor y serán accesibles para él siempre que entre a esa cuenta.
 - *Caso de simulación:* Archivo cuya extensión deberá ser *ongf*, *zip* y *rar*.
 - *Reemplazar / Mantener:* Si el archivo ya existe en el servidor con ese nombre y fue subido por el mismo usuario se le preguntará si quiere reemplazarlo.
- **Gestionar caso:** Una vez el caso esté simulado el usuario podrá realizar las siguientes acciones.
 - *Descargar:* El usuario descargará un archivo *ongf*, *rar* o *zip* en cuyo contenido estará el resultado de la simulación y los archivos del caso.
 - *Ver resultado:* Se mostrará una página con el resultado de la simulación.
 - *Compartir:* El usuario podrá dar acceso al caso a otros usuarios registrados en la aplicación.
 - *Gestionar permisos:* si el usuario con el que se quiere compartir existe en los registros se podrá decidir qué acciones se le permiten realizar sobre caso.
- **Borrar caso:** El usuario podrá borrar el caso del servidor una vez haya sido simulado.
- **Lanzar simulación:** El usuario podrá relanzar la simulación una vez haya sido simulado.

Con esta explicación y el diagrama se puede apreciar que existen dos módulos diferenciados en la aplicación, módulo de usuarios y de casos como ya se mencionó más arriba. Los usuarios no registrados podrán interactuar con ciertas funcionalidades del módulo de usuarios, mientras que los registrados podrán acceder a funcionalidades de los dos módulos.

La aplicación será accesible únicamente vía web y presentará una interfaz sencilla y amigable para facilitar a los usuarios su forma de interactuar con el front-end.

4.1.3. Identificación de los interesados en el sistema (Stakeholders)

En este apartado se identifican todos los grupos de interés en la aplicación para poder posteriormente determinar su participación, tanto en el estudio de la situación actual, como en la toma de requisitos de usuario, así como en el resto del proceso de desarrollo.

La siguiente tabla muestra los grupos de interés considerados para este proyecto:

Stakeholders	Descripción
--------------	-------------

Project Manager: Javier García Pérez	Persona responsable de la gestión y realización del proyecto.
Tutor: Israel González Carrasco	Persona encargada de supervisar el correcto desarrollo del proyecto
Usuario	Persona que usará la aplicación en el futuro

Tabla 1 Stakeholders

4.2. Estudio de la situación actual

Los siguientes apartados describirán la situación actual del entorno y hará un análisis de las aplicaciones que podrían ser competidoras del sistema a desarrollar, las cuales ya están implantadas en el mercado.

Con este estudio podremos determinar lo que queremos y no queremos a la hora de desarrollar la aplicación web, de esta forma se ahorrará tiempo en tener que deshacer funcionalidades ya implementadas que luego en el futuro no usará la aplicación.

Para finalizar, en este punto se hará una pequeña conclusión sobre qué características podrían mejorar las aplicaciones estudiadas y cuales han parecido interesantes para la futura elaboración de los requisitos.

4.2.1. Valoración del estudio de la situación actual

La aplicación a desarrollar es una app web que proporciona un repositorio al usuario para subir unos casos de simulación determinados. La aplicación permitirá lanzar contra un servidor ajeno la simulación del caso subido y tras la finalización de esta operación se proporcionará al usuario una serie de funcionalidades para interactuar con el caso (compartir, descargar, ver solución...), además se permitirá a este editar sus datos de registro (exceptuando el email).

No se ha encontrado un sistema parecido en internet (al menos conocido o que sea accesible para realizar un estudio), por lo que este estudio se centrará en la característica de repositorio online para analizar las aplicaciones competidoras.

En los siguientes párrafos se abordará el estudio de cuatro aplicaciones con cierta competitividad hacia las características del sistema a desarrollar. Finalmente se expondrá una tabla global con las principales características de cada una para así facilitar su comparación al lector[7].

- **Google Drive:**



Es un completo *office suite* de Google que proporciona almacenamiento en la nube. Tiene una oferta gratuita atractiva al proporcionar un espacio gratuito de 15 GB y una restricción de tamaño de archivo de 15 TB.

Además proporciona funcionalidades para compartir archivos y asignar permisos específicos a cada usuario, aunque estos permisos son algo limitados ya que únicamente se traducen a edición o lectura.

Por otro lado, una de sus grandes características es la opción de editar archivos almacenados (words, powerpoints,...) en tiempo real junto a otro usuario que tenga acceso al archivo.

También, aparte de ofrecer el servicio vía web, permite instalar la aplicación en los sistemas operativos Windows, Mac, Android, y iOS para así permitir su interacción de manera local.

Para finalizar, ofrece servicios de pago para aumentar el espacio de almacenamiento por si las necesidades del usuario o empresa así lo requirieran.



- **Dropbox:**

Es uno de los más populares del mundo en cuanto a almacenamiento en la nube debido a la facilidad en su uso al contar con una interfaz sencilla e intuitiva. Aunque su oferta gratuita no es muy atractiva, sólo ofrece 2 GB de espacio, se puede aumentar considerablemente por medio de acciones sin coste monetario.

Por otro lado, Dropbox ofrece tres tipos de cuentas: gratuita, pro y empresas. Cada una con unas características que se adaptan bien a los diferentes roles de los usuarios.

También ofrece la posibilidad de compartir los archivos y carpetas, aunque los permisos que se pueden asignar a los usuarios son algo pobres. Recientemente ha añadido la edición compartida de archivos a tiempo real.

Para finalizar también ofrece la posibilidad de descargar su versión escritorio en muchos sistemas operativos al igual que Drive.



- **OneDrive:**

Es el repositorio de almacenamiento online de Microsoft. Una de sus principales características es que su versión de escritorio viene integrado con los sistemas operativos Windows 8 y 8.1, permitiendo a usuarios noveles o con pocos conocimientos informáticos usarlo sin causarles quebraderos de cabeza.

Como oferta inicial gratuita ofrece 15 GB de espacio y 2 GB de tamaño máximo de archivo.

Por otro lado, al ser una aplicación de Microsoft, es la mejor manera de compartir y editar en grupo archivos de Office.

Para finalizar también cuenta con características de pago para ganar espacio, aunque ofrece maneras gratuitas para aumentarlo.



- **Box:**

Quizás la menos conocida y usada de las tres anteriores debido a su orientación más a empresas.

Es más completa en cuanto a la gestión de los permisos de compartición de archivos y carpetas.

No es la mejor opción para usuarios estándar debido a su pobre tamaño máximo de archivo en su versión gratuita, pero, como ya se mencionó en el primer párrafo, su orientación va más dirigida a empresas.

Para finalizar es compatible con aplicaciones de Salesforce y NetSuite, y existen plugins para serlo con Microsoft Office.

A continuación se muestra la tabla resumen con las principales características estudiadas y analizadas en los párrafos anteriores.

	Google Drive	Dropbox	OneDrive	Box
Tamaño máximo de archivo	5 TB	Ninguno	2 GB	250 MB(free) 5 GB
Espacio gratuito máximo	15 GB	2 GB	15 GB	10 GB
Espacio adicional gratuito	No	Sí	Sí	No
Versión de pago	Sí	Sí	Sí	Sí
Accesible	Sí	Sí	Sí	Sí
Compartir archivos	Sí	Sí	Sí	Sí
Edición compartida	Sí	Sí	Sí	No
Gestión de cuenta usuario	Sí	Sí	Sí	Sí

Tabla 2 Estudio de la situación actual

El objetivo principal del sistema a desarrollar no es el de actuar como repositorio online, debido a que únicamente se permite subir archivos muy específicos (casos de simulación) a un servidor y estos tras su subida podrán simularse contra un servidor ajeno. Pero estas aplicaciones son lo más parecido del mercado que se ha encontrado y ciertas características que ofrecen (tamaño máximo de archivo, sistemas de compartición,...) son útiles conocerlas para la realización de los requisitos y el desarrollo posterior de la aplicación.

4.3. Definición de los requisitos del sistema

En este punto, en base a la solicitud del cliente y el estudio de las aplicaciones competidoras, se van a extraer los requisitos del sistema y los que se deben recoger por parte del usuario. Cada requisito que sea añadido a la lista puede sufrir modificaciones según se avance el proyecto ya que hay que garantizar que sean correctos y atiendan a las necesidades del cliente.

Lo requisitos estarán correctamente definidos y claros, proporcionarán una visión general de la aplicación y serán la base del desarrollo del sistema. Con esta primera toma de requisitos se pretende orientar al cliente y así conseguir que ambas partes tengan una idea clara y completa de lo que quieren que sea la aplicación.

4.3.1. Identificación de requisitos

Para la elaboración de los requisitos de usuario se usarán tres tipos:

- **Requisitos de capacidad:** representan lo que necesitan los usuarios para resolver un problema o lograr un objetivo.
- **Requisitos de restricción:** son las restricciones impuestas por los usuarios sobre cómo se debe resolver el problema o cómo se debe alcanzar el objetivo.
- **Requisitos inversos:** recogen todos los puntos que la aplicación no debería hacer bajo ninguna circunstancia.

Cada requisito será expresado en una tabla que contendrá los siguientes campos con sus características:

- **Identificador:** Cada requisito tendrá un identificador único que seguirá la nomenclatura **UR – XYZ**, donde:
 - **UR:** User Requirements
 - **X:** Categoría del requisito (C,R o I)
 - **YZ:** numeración entera del requisito que crecerá de manera ascendente aumentando en una unidad.
- **Fuente:** Indica que parte de los interesados ha identificado el requisito, si ha sido el cliente o el Project manager.
- **Título:** Nombre identificativo del requisito.
- **Necesidad:** Este campo indicará con una escala de tres valores la necesidad de que ese requisito este implementado. Los tres valores a tomar son los siguientes:
 - **Esencial:** dicho requisito es necesario para el funcionamiento correcto de la aplicación.
 - **Deseable:** la implementación de ese requisito es un objetivo pero no necesario para una versión inicial.
 - **Opcional:** no influyen en el funcionamiento del proyecto y su realización no es importante.
- **Prioridad:** Todos los requisitos de usuario están sujetos a una prioridad para ayudar al desarrollador en su planificación. Al igual que la necesidad toma tres valores, los cuales son: alta, media y baja.
- **Verificabilidad:** Cada requisito de usuario será verificable. Esto significa que debe ser posible comprobar fehacientemente que el requisito se ha incorporado en el diseño, es decir, que se puede probar que el software aplica el requisito. Los valores que puede tomar son: alta, media y baja.
- **Estabilidad:** Algunos requisitos de usuario pueden permanecer estables durante toda la vida esperada del software; otros pueden ser más dependientes del feedback que se obtenga en los requisitos software, o el diseño.
- **Descripción:** breve texto que indique claramente el propósito del requisito.

El ejemplo de tabla que se usará a continuación en la lista de requisitos es la siguiente:

UR-XYZ			
Título	Ejemplo de requisito	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	El sistema debe...		

Tabla 3 Plantilla requisitos de usuario

4.3.1.1. Requisitos de capacidad

En este punto se recogen todos los requisitos de capacidad en base a lo expuesto por el cliente y las consideraciones del Project manager.

UR-C00			
Título	Identificación de usuario	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Para acceder a la aplicación completa el usuario deberá loguearse con sus siguientes datos de registro: <ul style="list-style-type: none"> • E-mail • Contraseña 		

Tabla 4 UR-C00

UR-C01			
Título	Recuperar contraseña	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	En caso de que el usuario no recuerde la contraseña se le permitirá recuperarla indicando su e-mail de registro		

Tabla 5 UR-C01

UR-C02			
Título	Registro de usuario	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Los usuarios no registrados podrán hacerlo introduciendo los siguientes datos: <ul style="list-style-type: none"> • E-mail • Nombre • Apellidos • Contraseña 		

Tabla 6 UR-C02

UR-C03			
Título	Activar cuenta	Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema

Descripción	Todo usuario tras registrarse deberá activar su cuenta para poder loguearse
--------------------	---

Tabla 7 UR-C03

UR-C04			
Título	Página tras login	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Tras loguearse satisfactoriamente el usuario será redirigido a su repositorio de casos, en caso contrario se le indicará que el login fue incorrecto		

Tabla 8 UR-C04

UR-C05			
Título	Apariencia repositorio	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	En el repositorio de cada usuario deberá mostrarse en la parte superior los siguientes datos: <ul style="list-style-type: none"> • Nombre de usuario • Indicador del estado del servidor de simulación de casos • Enlace a logout para cerrar sesión • Enlace a gestión de cuenta de usuario 		

Tabla 9 UR-C05

UR-C06			
Título	Sesión de usuario	Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Susceptible de cambios futuros
Descripción	Cuando un usuario inicie sesión, se cerrarán automáticamente las sesiones que estuvieran abiertas con anterioridad en ordenadores distintos		

Tabla 10 UR-C06

UR-C07			
Título	Casos de simulación	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	El usuario logueado podrá ver un listado con todos los casos disponibles en su repositorio, tanto propios como compartidos		

Tabla 11 UR-C07

UR-C08			
Título	Información de caso	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema

Descripción	Cada caso del repositorio del usuario mostrará la siguiente información:		
	<ul style="list-style-type: none"> • Propietario del caso (obligatorio en casos compartidos, en esta situación será el usuario que lo compartió) • Estado (<i>pendiente, calculando, simulado o errores</i>) • Fecha de última modificación • Enlace a mostrar resultados • Enlace a volver a lanzar simulación • Enlace a descarga del caso • Enlace a compartir caso • Enlace a eliminación del caso 		

Tabla 12 UR-C08

UR-C09			
Título	Compartir casos	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Una vez simulado el caso el usuario podrá compartirlo con usuarios registrados administrando los permisos que concede a cada uno		

Tabla 13 UR-C09

UR-C10			
Título	Subida de casos	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	El usuario podrá subir archivos de manera tradicional o usando la técnica de Drag&Drop		

Tabla 14 UR-C10

UR-C11			
Título	Ver resultado	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	El usuario podrá ver el resultado de un caso tras acabar su simulación		

Tabla 15 UR-C11

UR-C12			
Título	Re-lanzar simulación	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	El usuario podrá volver a simular un caso ya simulado		

Tabla 16 UR-C12

UR-C13			
Título	Descargar caso	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	El usuario podrá descargar un caso ya simulado		

Tabla 17 UR-C13

UR-C14			
Título	Borrar caso	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	El usuario podrá borrar un caso ya simulado		

Tabla 18 UR-C14

UR-C15			
Título	Confirmación de borrado	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Antes de borrar un caso se le pedirá al usuario confirmación		

Tabla 19 UR-C15

UR-C16			
Título	Idioma aplicación	Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	El usuario podrá elegir entre el inglés y español como idioma de la aplicación		

Tabla 20 UR-C16

UR-C17			
Título	Casos repetidos	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Si un usuario sube un caso con un nombre que ya existe en su repositorio se le dará la opción de machacar el antiguo o mantener las dos versiones		

Tabla 21 UR-C17

UR-C18			
Título	Estado simulación	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	El usuario podrá ver en tiempo real el progreso y estado de la simulación de sus casos		

Tabla 22 UR-C18

4.3.1.2. Requisitos de restricción

En este punto se recogen todos los requisitos de restricción considerados para el desarrollo del sistema.

UR-R00			
Título	Posición de las interfaces	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Las interfaces de usuario principales aparecerán en una posición centrada en la pantalla		

Tabla 23 UR-R00

UR-R01			
Título	Posición del menú	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	El menú de usuario aparecerá siempre en la parte superior de la pantalla		

Tabla 24 UR-R01

UR-R02			
Título	Estilo de las interfaces	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Susceptible a cambios futuros
Descripción	La apariencia de las interfaces de usuario deberá seguir un patrón (colores, fuentes, posición, ...) en todas las páginas de la aplicación		

Tabla 25 UR-R02

UR-R03			
Título	Formato de casos	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Susceptible a cambios futuros
Descripción	El usuario solo podrá subir al repositorio casos de simulación cuyo formato sea <i>ONGF</i> , <i>ZIP</i> o <i>RAR</i>		

Tabla 26 UR-R03

UR-R04			
Título	Permisos de casos	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Los usuarios con acceso a casos compartidos sólo podrán realizar las acciones permitidas por el usuario que lo compartió		

Tabla 27 UR-R04

UR-R05			
--------	--	--	--

Título	Asignar permisos shared	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Los usuarios con permisos de compartir un caso del cual no son propietarios sólo podrán asignar como máximo los permisos que ellos tienen sobre ese mismo caso a otros usuarios por motivos de seguridad		

Tabla 28 UR-R05

UR-R06			
Título	Carga de casos	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Susceptible a cambios futuros
Descripción	Los casos de simulación serán subidos al servidor mediante el método POST de http		

Tabla 29 UR-R06

UR-R07			
Título	Tamaño de casos	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Susceptible a cambios futuros
Descripción	El tamaño máximo del archivo del caso estará controlado y será establecido en el diseño del sistema		

Tabla 30 UR-R07

UR-R08			
Título	Retirada de permisos	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Al eliminar la compartición de un caso con un usuario se eliminarán todas las comparticiones que hizo este usuario con otro sobre el caso		

Tabla 31 UR-R08

UR-R09			
Título	Contenido caso	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Susceptible a cambios futuros
Descripción	Para que un caso pueda ser subido deberá contener un manifest.xml que sea validado frente a un esquema XSD		

Tabla 32 UR-R09

UR-R10			
Título	Elegir idioma	Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager

Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	La aplicación deberá permitir cambiar el idioma en cualquier página		

Tabla 33 UR-R10

UR-R11			
Título	Inicio de sesión	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Cuando un usuario se loguea se descomprime el contenido HTML de los casos en una carpeta temporal asociada a ese usuario		

Tabla 34 UR-R11

UR-R12			
Título	Cierre de sesión	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Cuando un usuario cierra sesión se elimina la carpeta temporal asociada con todos el contenido HTML de sus casos		

Tabla 35 UR-R12

UR-R13			
Título	Comunicación servidor	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	El sistema de la aplicación web y el servidor de simulaciones se comunicarán mediante WebServices		

Tabla 36 UR-R13

UR-R14			
Título	Heartbeat	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	El sistema de la aplicación web y el servidor de simulaciones se comprobarán mutuamente su estado mediante un sistema de HeartBeat		

Tabla 37 UR-R14

UR-R15			
Título	Casos borrados	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Los caos que hayan sido borrados se almacenarán en una carpeta aparte en el servidor		

Tabla 38 UR-R15

UR-R16			
Título	Almacenamiento	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	La información de los casos y de los usuarios será almacenada en una base de datos		

Tabla 39 UR-R16

UR-R17			
Título	Casos en servidor	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Los casos de los usuarios permanecerán comprimidos en el servidor, sólo se descomprimirán tras login en carpetas temporales (UR-R11), cuyo contenido será borrado tras el cierre de sesión (UR-R12)		

Tabla 40 UR-R17

UR-R18			
Título	Página siguiente y anterior	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	El sistema debe permitir al usuario cambiar de página de vista en el caso de que tenga tantos casos de simulación subidos que no puedan ser mostrados en una sola ventana		

Tabla 41 UR-R18

4.3.1.3. Requisitos inversos

A continuación se muestran los requisitos inversos de nuestra aplicación, estos recogen todos los puntos que la aplicación no debería hacer bajo ninguna circunstancia. Pueden ser infinitos, pero se mostraran los más relevantes.

UR-I00			
Título	Protección de casos	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Ningún usuario tendrá acceso a la información y manipulación de casos que no son suyos o no le han sido compartidos		

Tabla 42 UR-I00

UR-I01			
Título	Publicidad	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager

Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	La aplicación no mostrará ningún tipo de publicidad y anuncios mientras se interactúa con ella		

Tabla 43 UR-I01

UR-I02			
Título	Cambiar email	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Ningún usuario podrá cambiar su email de registro		

Tabla 44 UR-I02

UR-I03			
Título	Contraseñas aplicación	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Nunca se mostrarán en claro en la aplicación las contraseñas de los usuarios		

Tabla 45 UR-I03

4.4. Estudio de alternativas a la solución

Una vez estudiada la situación actual y extraídos los requisitos de usuario ya se puede valorar las diferentes soluciones que se pueden usar para desarrollar el sistema.

Las alternativas estudiadas se valorarán de manera cualitativa y cuantitativa en función a los parámetros que se consideren oportunos.

En los siguientes puntos se exponen las alternativas estudiadas para la realización de este TFG.

4.4.1. Alternativa 1: Java Servlets + MySQL

Basada en las especificaciones de Java EE permite mostrar y realizar un contenido dinámico para la aplicación web. Con los Servlets extenderemos las funcionalidades que proporcionará el servidor y con las páginas HTML y JSP mostraremos el contenido web.

La tecnología Java Servlets[10] permite controlar de una manera sencilla todas las peticiones que llegan al servidor mediante el uso de los Servlets o filtros, incluso se puede extender la funcionalidad de los servlets encapsulándola en EJB (Enterprise JavaBeans). Todos los datos que el servlet procese son luego volcados sobre páginas JSP que muestran al usuario la información de una manera dinámica. Un ejemplo de este funcionamiento se muestra en el siguiente esquema:

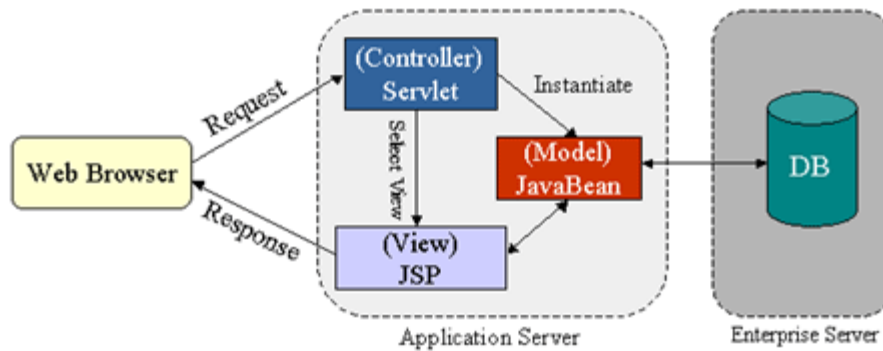


Ilustración 6 Arquitectura Servlet | Recuperado de: <https://knowthediff.wordpress.com/2014/09/30/servlets-vs-cgi>

Por otro lado se utilizará MySQL para la gestión de datos por ser gratuito. Este gestor será el considerado también en las otras alternativas ya que es el más simple y el que más se adapta para el proyecto, para esta propuesta la API de JDBC proporciona un conjunto de librerías perfectas para la interacción entre los servlets y la base de datos.

Se han desechado otras alternativas como Oracle o MongoDB ya que son más complejos y aunque en verdad tienen más funcionalidades y son más completos, la mayoría de sus características no nos servirían para el sistema o no aportarían algo que con MySQL no se pueda también conseguir. Además, el caso que se desee trabajar con objetos, cosa que podría aportar MongoDB, la API de JPA (Java Persistence API[2]) permite trabajar con objetos en el programa aunque luego por detrás se esté usando una base de datos relacional.

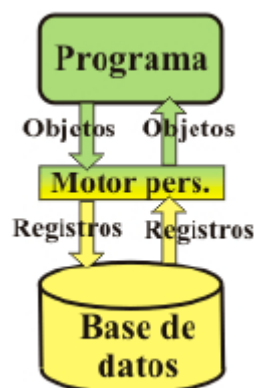


Ilustración 7 Modelo de persistencia | Recuperado de: <http://www.ufg.edu.sv/ufg/theorethikos/Julio04/mdp6.html>

Por último se muestra una tabla resumen con las principales características de esta alternativa analizada en este apartado.

Alternativa 1: Java Servlets + MySQL	
Interfaz de Usuario	Páginas HTML/JSP
Lógica de negocio	Servlets/Filtros/EJB
Gestión de datos	MySQL
Servidor	Apache Tomcat/JBoss/Glassfish/IBM Bluemix

Ilustración 8 Resumen alternativa 1

4.4.2. Alternativa 2: PHP + MySQL

PHP (Hypertext Preprocessor) es un lenguaje que se incrusta fácilmente en el HTML de las páginas y permite el desarrollo de páginas web dinámicas.

El código PHP es ejecutado en el servidor generando el HTML resultante que es enviado al cliente. Una de las cosas más atractivas de este lenguaje es su simplicidad para personas con poco conocimiento en esta área y a que es un lenguaje de tipado débil.

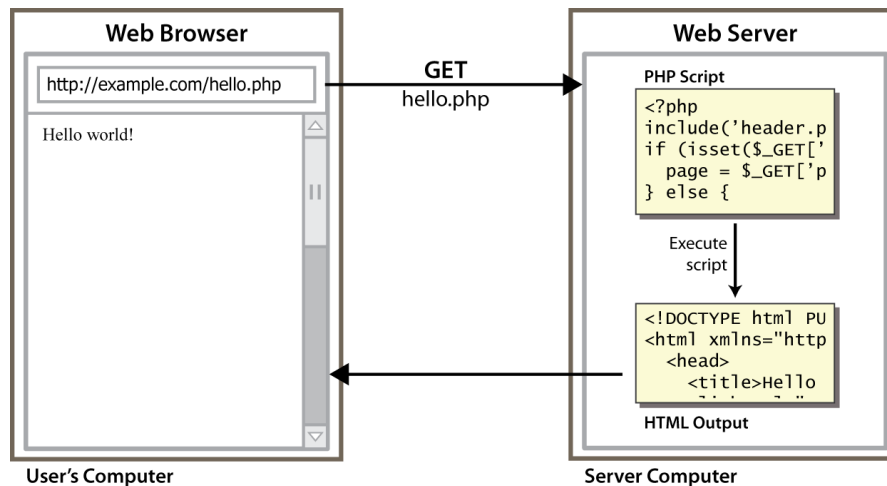


Ilustración 9 Arquitectura PHP | Recuperado de: <http://www.webstepbook.com/supplements-2ed/slides/chapter05-php.shtml>

Por último, PHP es considerado un lenguaje más 'económico' debido a que es un lenguaje de código abierto y a que funciona universalmente en cualquier plataforma.

Las razones de usar MySQL son las mismas que las expuestas en el apartado anterior, aunque si se usa PHP tendremos menos API que puedan ayudar en la iteración con la base de datos.

Por último se muestra una tabla resumen con las principales características de esta alternativa analizada en este apartado.

Alternativa 2: PHP + MySQL	
Interfaz de Usuario	Páginas HTML + PHP
Lógica de negocio	Script PHP
Gestión de datos	MySQL
Servidor	Apache

4.5. Valoración de las alternativas

Tras haber estudiado estas dos posibles alternativas a la solución se procederá a elegir la adecuada para el desarrollo del TFG.

Se hará un análisis de sus ventajas y posteriormente se realizará una elección en base a parámetros como fiabilidad, coste, soporte, experiencia,...

4.5.1. Valoración de alternativa 1: Java Servlets + MySQL

Las ventajas más destacadas para la alternativa 1: *Java Servlets + MySQL* son las recogidas en los siguientes puntos:

- Utilización de recursos mínima por parte de los Servlets. Cada petición se ejecuta en un hilo y no en un proceso.
- Gran documentación y guías de Java.
- Los Servlets son portables al poder usarse en la mayoría de sistemas operativos y servidores web.
- Los Servlets son integrables con páginas JSP y filtros.
- Por ser Java ofrece: máquina virtual, chequeo de tipos, gestión de memoria, excepciones, seguridad, etc.

4.5.2. Valoración de alternativa 2: PHP + MySQL

Las ventajas más destacadas para la alternativa 2: *PHP + MySQL* son las recogidas en los siguientes puntos:

- Lenguaje multiplataforma.
- La mayoría de servidores cuentan con integración para PHP por lo que facilita su despliegue.
- Lenguaje libre y abierto.
- Poca necesidad de experiencia.
- Gran simplicidad en el desarrollo de código.

4.6. Selección de la solución

Para la elección de la mejor alternativa, a parte de las ventajas ya mencionadas, se va a usar la valoración una serie de parámetros a los que se les asignará una ponderación diferente en base a la importancia para nuestro sistema.

En las tablas y gráficos siguientes se muestra esta valoración y posteriormente explicará cual ha sido la alternativa elegida.

- **Alternativa 1: *Java Servlets + MySQL***

Criterio	Valoración	Criterio ponderación	Valoración ponderada
Coste	10	9	90
Fiabilidad	8	10	80
Soporte	6	8	48
Experiencia	6	6	36
Instalación	5	4	20
Referencias	8	7	56

Mantenibilidad	9	8	72
Seguridad	9	9	81
Escalabilidad	8	7	56
Rendimiento	8	10	80
Media	7,7		61,9
	Resultado final		6,2

Tabla 46 Estudio alternativa 1

- **Alternativa 2: PHP + MySQL**

Criterio	Valoración	Criterio ponderación	Valoración ponderada
Coste	10	9	90
Fiabilidad	6	10	60
Soporte	4	8	32
Experiencia	8	6	48
Instalación	8	4	32
Referencias	8	7	56
Mantenibilidad	7	8	56
Seguridad	7	9	63
Escalabilidad	5	7	35
Rendimiento	8	10	80
Media	7,1		55,2
	Resultado final		5,5

Tabla 47 Estudio alternativa 2

Para la comprensión de estos resultados nos apoyaremos en gráficos sencillos que mostrarán desde diferentes ángulos los resultados de las tablas de arriba.

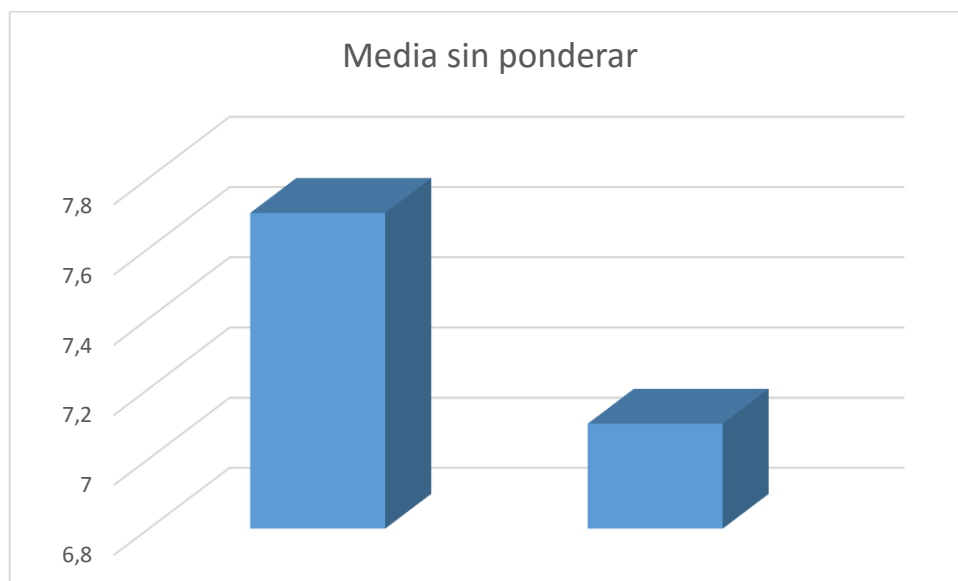


Ilustración 10 Alternativas: Media sin ponderar

En este gráfico se muestra la comparación gráfica de las medias de la alternativa 1 (izquierda) y la alternativa 2 (derecha) donde a priori parece mejor la alternativa 1.

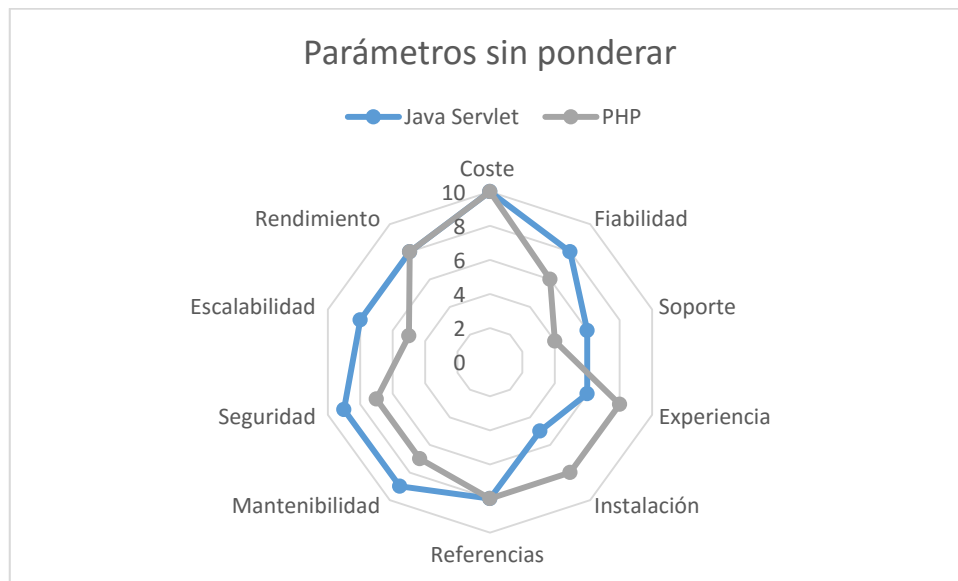


Ilustración 11 Alternativas: Parámetros sin ponderar

En el siguiente gráfico analizamos las alternativas en base a los parámetros de las tablas. Se puede apreciar una gran igualdad, pero la alternativa 1 toma ventaja en características más importantes como son la escalabilidad, seguridad y mantenibilidad. Mientras que la alternativa 2 es mejor en experiencia e instalación, pero a estos dos parámetros no se les da tanta ponderación como a los otros.

Por último en el siguiente gráfico se muestra la comparación con los resultados ya ponderados (izquierda alternativa 1 y derecha alternativa 2).

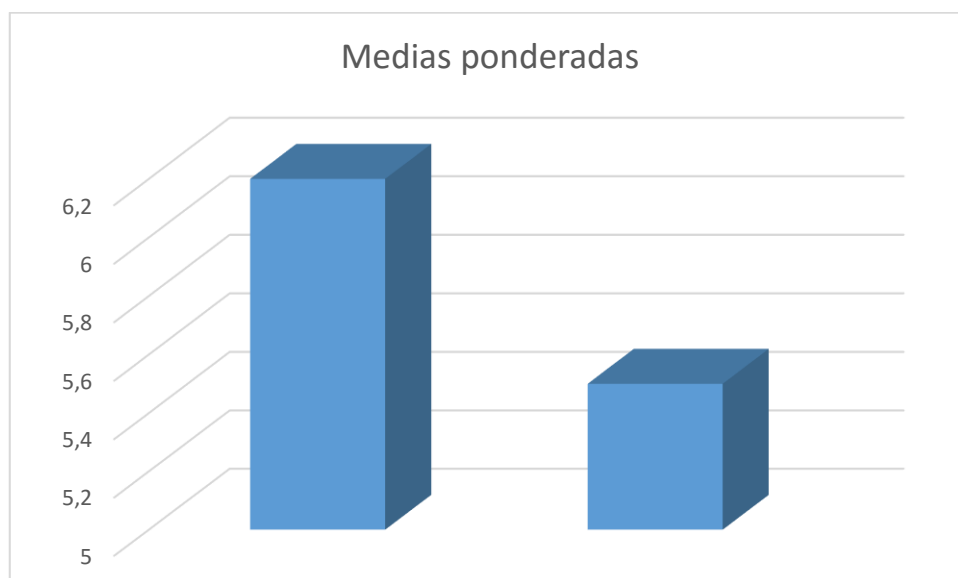


Ilustración 12 Alternativas: Medias ponderadas

En definitiva, una vez hecho este análisis de las alternativas posibles la mejor solución para el desarrollo del TFG es Java Servlets + MySQL, aunque la alternativa de PHP no hubiera sido

tampoco mala solución debido a que se diferencian en pocos aspectos para lo que se quiere hacer.

5. Gestión del proyecto

Este apartado tiene como finalidad principal la planificación, el seguimiento y el control de las actividades y de los recursos humanos y materiales que intervienen en el desarrollo del sistema, para evitar desviaciones de costes, duración o funcionalidad del proyecto. Con este control se persigue conocer en todo momento qué problemas se producen y resolverlos o paliarlos a la mayor brevedad para minimizar sus consecuencias.

5.1. Ciclo de vida

En esta sección se define el porqué del ciclo de vida elegido y se explicará cada una de las fases del mismo.

La estrategia que se ha elegido ha dependido de las funciones características del proyecto a realizar, tales como su tamaño o su complejidad.

Por otro lado, las funciones principales que deben quedar claras antes de elegir un ciclo de vida software son:

- Determinar el orden de las fases y procesos involucrados en el desarrollo del software y su evolución incluyendo la explotación y el mantenimiento.
- Establecer los criterios de transición para pasar de una fase a la siguiente (productos intermedios). Todo ello incluye los criterios para verificar la terminación de la fase actual y los criterios para seleccionar e iniciar la fase siguiente.

Una vez hecho este análisis se deberá elegir entre los ciclos de vida software que más se adapten al proyecto, los cuales pueden ser:

- Clásico o en cascada: modelo de etapas que se procesa de un modo lineal y en la que un proceso no puede comenzar hasta que se ha terminado el anterior.
- Incremental: el desarrollo del sistema se divide en subsistemas independientes, con estructura y organización diferentes.
- En espiral: las actividades de este modelo se conforman en una espiral, en la que cada bucle o iteración representa un conjunto de actividades. Las actividades no están fijadas a ninguna prioridad, sino que las siguientes se eligen en función del análisis de riesgo, comenzando por el bucle interior.
- Desarrollo ágil: utiliza un desarrollo iterativo como base para abogar por un punto de vista más ligero y más centrado en las personas que en el caso de las soluciones tradicionales. Los procesos ágiles utilizan retroalimentación en lugar de planificación, como principal mecanismo de control.
- ...

En el caso de este TFG se ha elegido el ciclo en cascada o clásico debido a que las necesidades (requisitos) del usuario están bien entendidas y documentadas, no existen módulos independientes que obliguen a dividir o tratar el sistema de manera diferente y además a que no se realizarán entregas intermedias con una funcionalidad parcial del sistema.

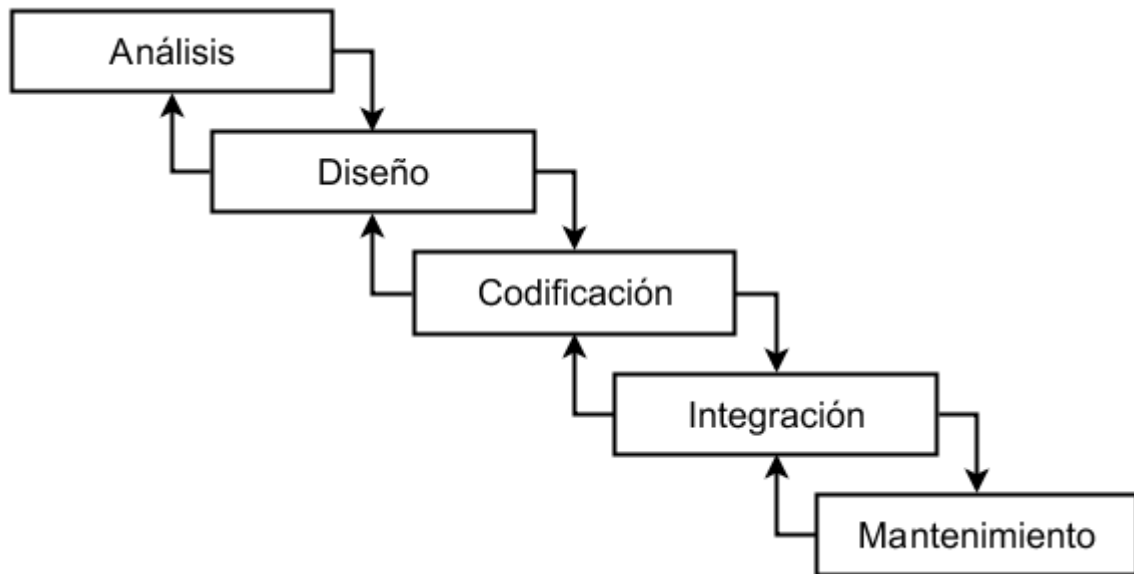


Ilustración 13 Diagrama ejemplo de ciclo de vida en cascada

A continuación se pasan a detallar las fases de este ciclo de vida software:

- Determinar la viabilidad: aunque no se recoge esta etapa en la imagen de ejemplo de arriba, en esta fase se determina la viabilidad del sistema en base a datos recopilados y analizados desde una perspectiva técnica, económica y operativa. Además en esta fase se redactan los requisitos de usuario.
- Análisis: en esta etapa se formaliza el dominio del problema y se hace una primera definición de las funcionalidades de la aplicación a desarrollar.
- Diseño: en esta etapa se recoge al nivel más bajo las definiciones realizadas en el análisis anterior. La redacción de esta fase debe quedar de una manera en la que los desarrolladores de la aplicación tengan lo suficientemente claro el trabajo a realizar.
- Codificación o desarrollo: en esta etapa se elabora toda la lógica de la aplicación. Sus clases, modelo de datos, capa de presentación, ...
- Integración: en esta etapa, como su nombre indica, se integran todas las capas del proyecto y se verifica su comportamiento y cumplimiento de los requisitos por medio de pruebas.
- Entrega y mantenimiento: etapa en la que se proporciona el resultado al cliente realizando la instalación del sistema. Posteriormente, si estaba incluido en el contrato, se realizará un proceso de mantenimiento periódico.

5.2. Organización del proyecto

Todo proyecto software necesita de una planificación clara y detallada en el que se informe del tiempo estimado de duración de las actividades y tareas planificadas para la realización del proyecto, así como de los recursos asignados a cada una de ellas.

La planificación desarrollada para este TFG viene en función del ciclo de vida en cascada elegido.

En la siguiente imagen se muestra la descomposición de las etapas de este ciclo en sus respectivas actividades y tareas dentro del proyecto.

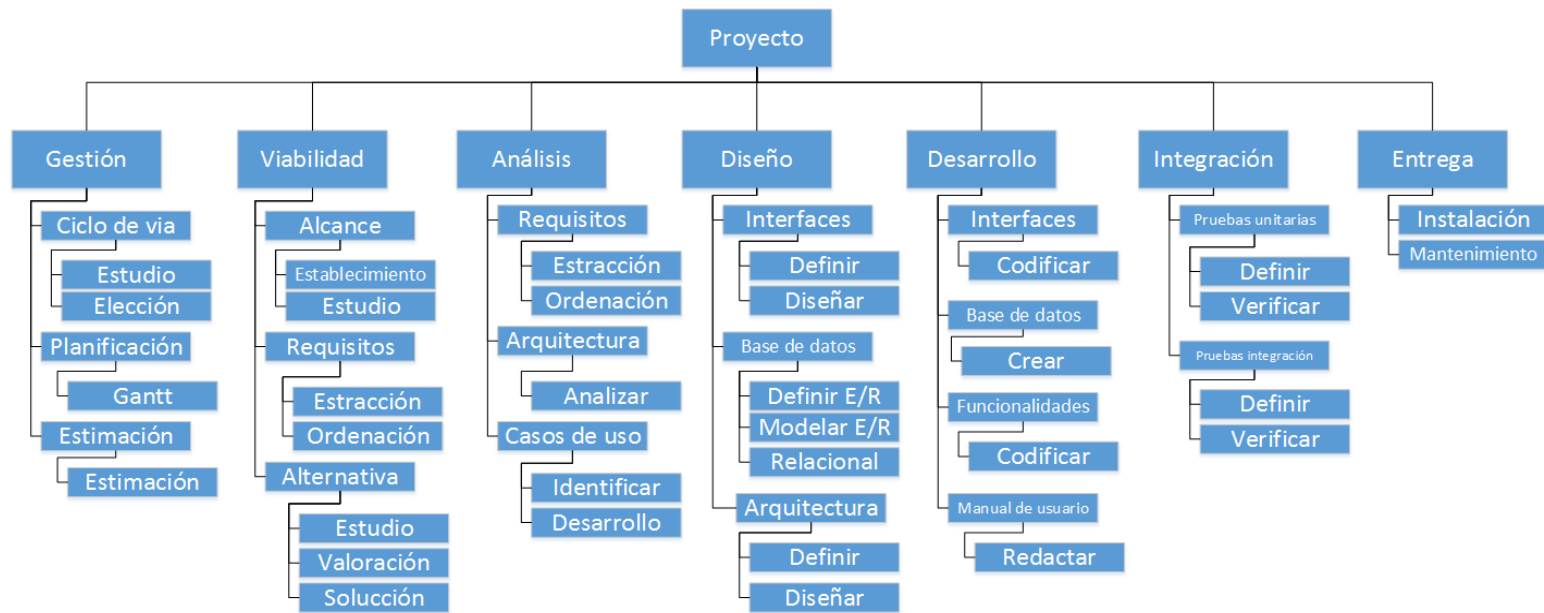


Ilustración 14 Descomposición de tareas

Tal y como se muestra en la figura anterior, el proyecto sobre el que trata este documento queda dividido en las siguientes fases:

- **Gestión del proyecto:** fase en la que se realiza la planificación y estimación de las actividades y tareas del proyecto, así como la elección del ciclo de vida a seguir para la realización de las siguientes fases.
- **Determinar viabilidad:** fase en la que se recogen los requisitos de usuario en base a los datos recopilados. En referencia a estos requisitos se estudia, analiza y elige una alternativa para la solución del sistema si se considera este viable para su realización.
- **Análisis del sistema:** fase en la que se identifican los requisitos software, se realiza a alto nivel una definición de la arquitectura a seguir y finalmente recoge diferentes casos de uso en base a los requisitos.
- **Diseño del sistema:** en esta fase se definen las GUI a usar en el sistema, se estudia el modelo de la base datos a utilizar y finalmente se realiza una descripción a bajo nivel de la estructura y diseño del proyecto.
- **Desarrollo del sistema:** básicamente se centra en la codificación de todos los aspectos recogidos en el análisis y posteriormente ampliados en el diseño (interfaces, funcionalidades, base de datos,...). Finalmente se completa con la redacción del manual de usuario.
- **Integración del sistema:** consiste en la verificación del correcto y completo funcionamiento del sistema mediante el uso de pruebas.
- **Entrega:** Entrega e instalación del sistema completo al cliente

5.3. Planificación

Uno de los aspectos más importantes de todo proyecto software es realizar una planificación correcta y adecuada de las tareas para no incumplir en plazos.

Para la planificación del TFG se expondrá un diagrama GANTT a continuación con todas las actividades y tareas del proyecto. En este diagrama se podrá ver en detalle la duración total de estas y del proyecto en conjunto.

Posteriormente se mostrará una tabla más concreta con la información de inicio, fin y duración de cada tarea.

A continuación se adjunta el diagrama GANTT con las actividades y tareas del proyecto a desarrollar. La duración está medida en días laborables y se ha tomado como fecha inicial el 12/01/2015 y como fecha final el 12/06/2015.

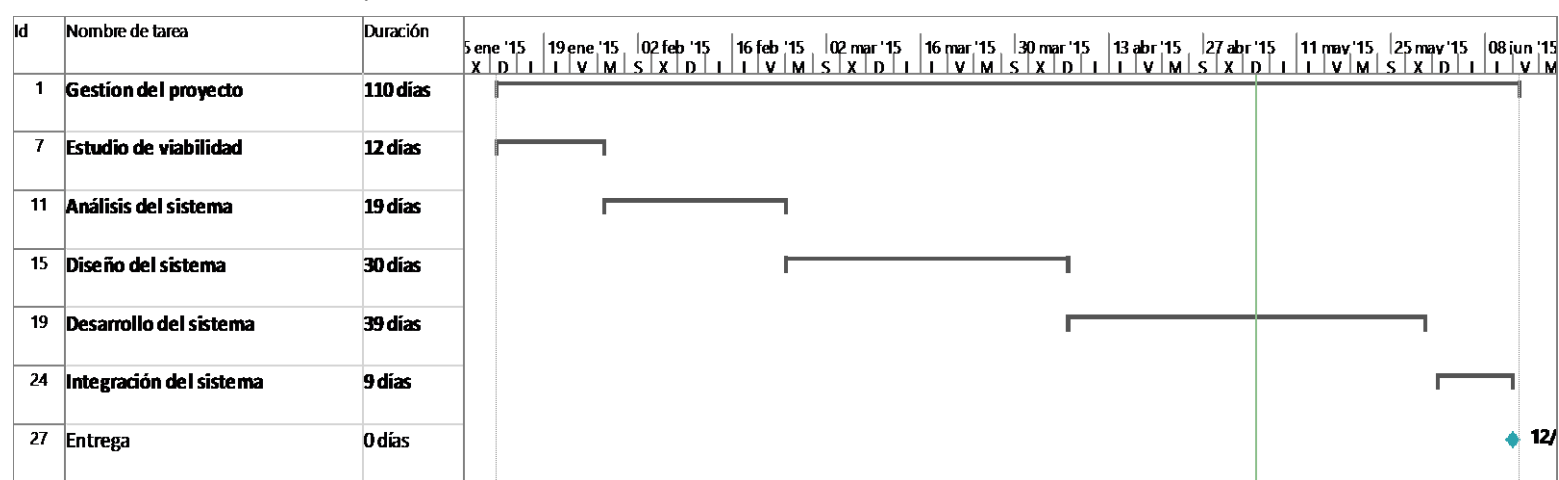


Ilustración 15 Diagrama GANTT (resumido)

En el diagrama anterior se muestran las seis tareas principales consideradas para este TFG, y finalmente se marca como un hito la entrega, cuya fecha se ha establecido al 12/06/2015.

A continuación se muestra un desglose de cada actividad de proyecto con sus respectivas tareas.

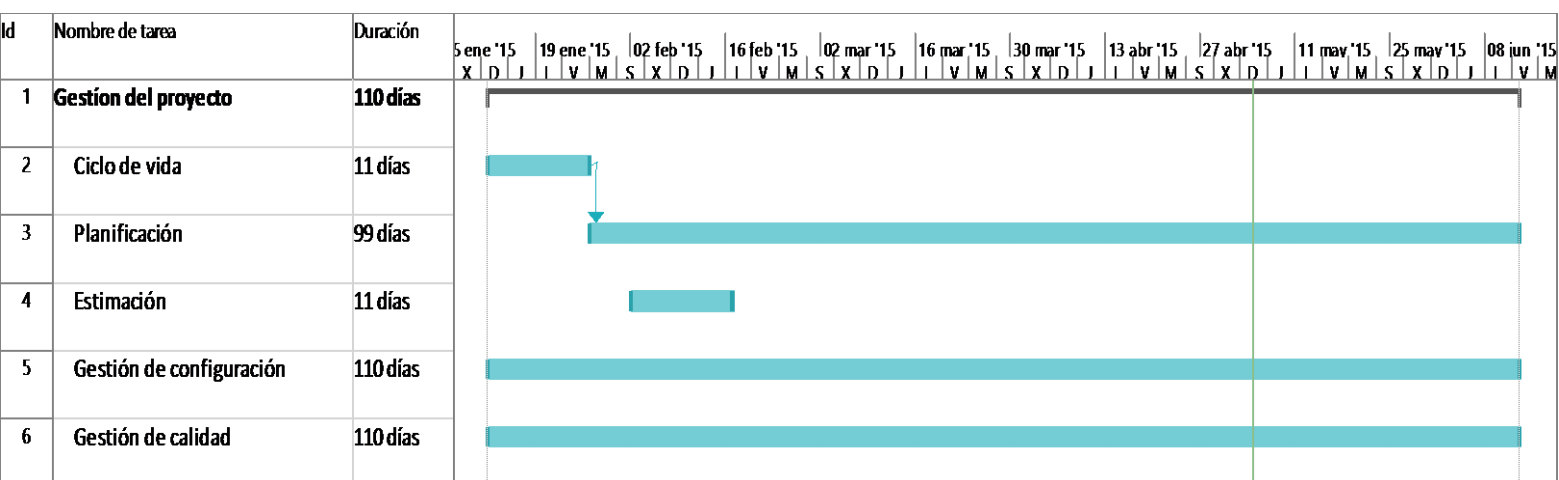


Ilustración 16 GANTT Gestión de proyecto

Id	Nombre de tarea	Duración	Comienzo	5 ene '15	19 ene '15	
7	Estudio de viabilidad	12 días	lun 12/01/15	X	D	J
8	Determinar alcance	2 días	lun 12/01/15			
9	Requisitos de usuario	3 días	mié 14/01/15			
10	Estudio y elección de alternativa	7 días	lun 19/01/15			

Ilustración 17 GANTT Estudio de viabilidad

Id	Nombre de tarea	Duración	Comienzo	ene '15	02 feb '15	16 feb '15	
11	Análisis del sistema	19 días	mié 28/01/15	V	M	S	X
12	Casos de uso	8 días	mié 28/01/15				
13	Requisitos software	11 días	mié 28/01/15				
14	Analizar arquitectura	8 días	jue 12/02/15				

Ilustración 18 GANTT Análisis del sistema

Id	Nombre de tarea	Duración	Comienzo	feb '15	02 mar '15	16 mar '15	30 mar '15	13
15	Diseño del sistema	30 días	mar 24/02/15	V	M	S	X	D
16	Interfaces	13 días	mar 24/02/15					
17	Definir y modelar BD	11 días	lun 02/03/15					
18	Definir y diseñar arquitectura	20 días	mar 10/03/15					

Ilustración 19 GANTT Diseño del sistema

Id	Nombre de tarea	Duración	Comienzo	abr '15	13 abr '15	27 abr '15	11 may '15	25 may '15
19	Desarrollo del sistema	39 días	mar 07/04/15	D	I	I	V	M
20	Codificar interfaces	11 días	mar 07/04/15					
21	Crear BD	8 días	mar 07/04/15					
22	Codificar funcionalidades	23 días	mié 22/04/15					
23	Manual de usuario	5 días	lun 25/05/15					

Ilustración 20 GANTT Desarrollo del sistema

Id	Nombre de tarea	Duración	Comienzo	5 may '15	08 jun '15
				X D J L V M	
24	Integración del sistema	9 días	lun 01/06/15		
25	Pruebas unitarias	9 días	lun 01/06/15		
26	Pruebas de integración	9 días	lun 01/06/15		

Ilustración 21 GANTT Integración del sistema

Id	Nombre de tarea	Duración	Comienzo	08 jun '15
				L V M S
27	Entrega	0 días	vie 12/06/15	12/06

Ilustración 22 GANTT Entrega

Tras estos diagramas se adjunta una tabla resumen con la duración, fecha inicio y fecha fin de cada una de las actividades y tareas de este proyecto.

Tarea	Duración	Comienzo	Fin
Gestión del proyecto	110 días	lun 12/01/15	vie 12/06/15
<i>Ciclo de vida</i>	<i>11 días</i>	<i>lun 12/01/15</i>	<i>lun 26/01/15</i>
<i>Planificación</i>	<i>99 días</i>	<i>mar 27/01/15</i>	<i>vie 12/06/15</i>
<i>Estimación</i>	<i>11 días</i>	<i>lun 02/02/15</i>	<i>lun 16/02/15</i>
<i>Gestión de configuración</i>	<i>110 días</i>	<i>lun 12/01/15</i>	<i>vie 12/06/15</i>
<i>Gestión de calidad</i>	<i>110 días</i>	<i>lun 12/01/15</i>	<i>vie 12/06/15</i>
Estudio de viabilidad	12 días	lun 12/01/15	mar 27/01/15
<i>Determinar alcance</i>	<i>2 días</i>	<i>lun 12/01/15</i>	<i>mar 13/01/15</i>
<i>Requisitos de usuario</i>	<i>3 días</i>	<i>mé 14/01/15</i>	<i>vie 16/01/15</i>
<i>Estudios y elección de alternativa</i>	<i>7 días</i>	<i>lun 19/01/15</i>	<i>mar 27/01/15</i>
Análisis del sistema	19 días	mié 28/01/15	lun 23/02/15
<i>Casos de uso</i>	<i>8 días</i>	<i>mié 28/01/15</i>	<i>vie 06/02/15</i>
<i>Requisitos software</i>	<i>11 días</i>	<i>mié 28/01/15</i>	<i>mié 11/02/15</i>
<i>Analizar arquitectura</i>	<i>8 días</i>	<i>jue 12/02/15</i>	<i>lun 23/02/15</i>
Diseño del sistema	30 días	mar 24/02/15	lun 06/04/15
<i>Interfaces</i>	<i>13 días</i>	<i>mar 24/02/15</i>	<i>jue 12/03/15</i>
<i>Definir y modelar BD</i>	<i>11 días</i>	<i>lun 02/03/15</i>	<i>lun 16/03/15</i>
<i>Definir y diseñar arquitectura</i>	<i>20 días</i>	<i>mar 10/03/15</i>	<i>lun 06/04/15</i>
Desarrollo del sistema	39 días	mar 07/04/15	vie 29/05/15
<i>Codificar interfaces</i>	<i>11 días</i>	<i>mar 07/04/15</i>	<i>mar 21/04/15</i>
<i>Crear BD</i>	<i>8 días</i>	<i>mar 07/04/15</i>	<i>jue 16/04/15</i>
<i>Codificar funcionalidades</i>	<i>23 días</i>	<i>mié 22/04/15</i>	<i>vie 22/05/15</i>
<i>Manual de usuario</i>	<i>5 días</i>	<i>lun 25/05/15</i>	<i>vie 29/05/15</i>
Integración del sistema	9 días	lun 01/06/15	jue 11/06/15
<i>Pruebas unitarias</i>	<i>9 días</i>	<i>lun 01/06/15</i>	<i>jue 11/06/15</i>
<i>Pruebas de integración</i>	<i>9 días</i>	<i>lun 01/06/15</i>	<i>jue 11/06/15</i>
Entrega	0 días	vie 12/06/15	vie 12/06/15

Tabla 48 Duración de actividades y tareas

5.4. Estimación de costes

Para realizar la estimación del coste del proyecto[4] se ha considerado un sueldo de 16 €/h para el Project manager, que será el que realice todas las actividades.

Como ya se mencionó en el apartado anterior sólo se consideran los días laborales y para cada jornada se le asignará 8 horas de trabajo.

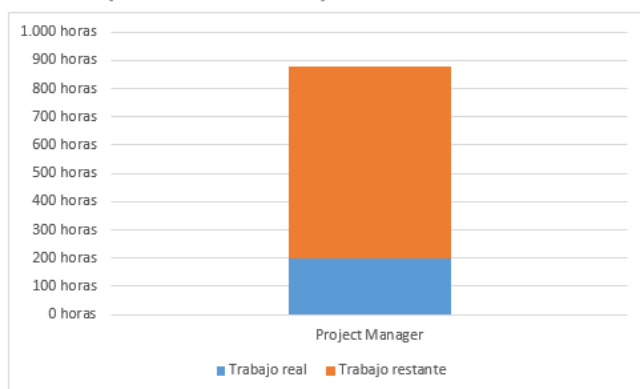
Para el cálculo total del coste no se ha tenido en cuenta ningún coste material ni adicional, exceptuando el IVA.

En este apartado se expondrá el total del coste del proyecto en general y posteriormente se expondrá el coste por actividad. Se usarán diagramas y tablas para ayudar a su correcta comprensión.

VISIÓN GENERAL DE LOS RECURSOS

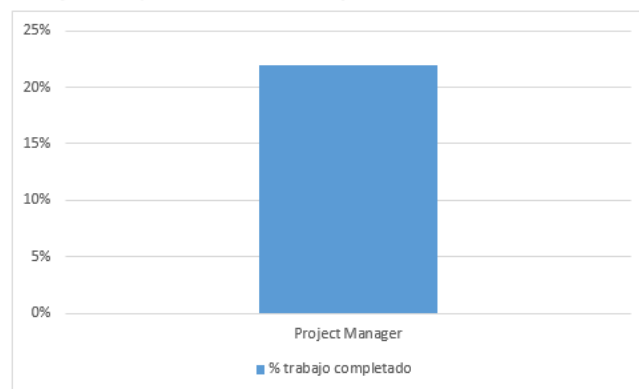
ESTADÍSTICAS DE RECURSOS

Estado de trabajo de todos los recursos de trabajo.



ESTADO DEL TRABAJO

% trabajo realizado por todos los recursos de trabajo.



ESTADO DE LOS RECURSOS

Resta trabajo para todos los recursos de trabajo

Nombre	Comienzo	Fin	Trabajo restante
Project Manager	lun 12/01/15	vie 12/06/15	678,97 horas

Ilustración 23 Estimación de recursos

En el gráfico anterior se puede apreciar las horas totales del proyecto y el progreso completado hasta este punto actual que incluye el estudio de viabilidad, elección del ciclo de vida, planificación,... y las horas restantes estimadas para finalizar.

Adentrándonos al terreno monetario el monetario el coste por actividad es el siguiente dependiendo de su duración:

Tarea	Duración	Comienzo	Fin	Coste
Gestión del proyecto	110 días	lun 12/01/15	vie 12/06/15	14,080.00 €
Estudio de viabilidad	12 días	lun 12/01/15	mar 27/01/15	1,536.00 €
Análisis del sistema	19 días	mié 28/01/15	lun 23/02/15	2,432.00 €
Diseño del sistema	30 días	mar 24/02/15	lun 06/04/15	3,840.00 €
Desarrollo del sistema	39 días	mar 07/04/15	vie 29/05/15	4,992.00 €
Integración del sistema	9 días	lun 01/06/15	jue 11/06/15	1,152.00 €
Entrega	0 días	vie 12/06/15	vie 12/06/15	0 €

Total (sin IVA): 28,032 €
IVA (18%): 5,045.76 €
Total (IVA incluido): 33,077.76 €

Tabla 49 Tabla de costes

El cálculo se ha realizado aplicando un coste de 16€ a cada hora de duración de las actividades, posteriormente al total se le ha sumado un IVA del 18%.

6. Análisis del sistema

Este punto recoge de forma concisa el análisis del sistema a desarrollar. Este apartado busca obtener una especificación detallada del sistema y servir como base centrar al diseño del mismo.

6.1. Objetivo

Como se ha mencionado en la introducción, el principal objetivo es obtener una especificación detallada del sistema que se va a construir. Posteriormente, este análisis servirá de base para el diseño del sistema.

En esta fase de análisis se pretende definir el problema a resolver. Se trata de captar las necesidades que debe resolver y modelar el problema utilizando distintas técnicas, en función de las características de este proyecto, para posteriormente en el diseño del sistema resolverlo.

6.2. Alcance

El alcance de este punto del documento es determinar las necesidades del cliente. Para ello se determinarán los límites del sistema identificando los participantes involucrados a lo largo de la vida del mismo y el entorno tecnológico.

Además se definirán los requisitos software que han de ser cumplidos por el sistema. Estos requisitos tomarán como base los requisitos de usuario redactados en el apartado 4.3. *Definición de los requisitos del sistema* del EVS.

6.3. Definición del sistema

Este apartado define la aplicación al completo, con todas las funcionalidades que debe cumplir y su alcance, la identificación del entorno tecnológico, los estándares y normas usadas, además de la identificación de los usuarios de la aplicación y del análisis.

6.3.1. Determinación del alcance del sistema

Este apartado determina el alcance del sistema a desarrollar con el fin de cumplir las necesidades planteadas por el cliente que se han recogido en el punto 4.1. *Establecimiento del alcance del sistema* del EVS.

Resumiendo lo recogido en el punto de ese apartado, el sistema consiste en una aplicación web que proporciona a los usuarios un repositorio para alojar casos de simulación y desde el cual los usuarios podrán gestionar estos. Permitiéndoles lanzar la simulación, compartir, descargar, ver el resultado,... de los casos en los que sean propietarios o les hayan sido compartidos. Además el usuario podrá recuperar y modificar sus datos de registro (excepto el email) una vez creada la cuenta.

Este trabajo será supervisado y guiado por Israel González Carrasco y realizado por Javier García Pérez, estudiante de la titulación de Ingeniería Informática en la Universidad Carlos III de Madrid.

6.3.2. Identificación del entorno tecnológico

Ver el apartado 6.3.3.3. *Entorno operacional* de este documento.

6.3.3. Especificación de estándares y normas

A lo largo de la realización de este TFG es necesario cumplir ciertos estándares, normas o leyes existentes para obtener un producto de calidad. Los siguientes apartados describen dichas normas, las restricciones generales de la aplicación y el entorno operacional.

Todo este documento está desarrollado bajo la metodología Métrica v3. Esto significa que las fases de las que consta el producto están basadas en este estándar, así mismo, no implica que la aplicación realice todas ellas, sino las necesarias para la correcta consecución del proyecto.

Otro estándar seguido en este proyecto es UML, el cual será usado en la realización del diagrama de clases y otros diagramas menores que pueda recoger el documento.

Por último, se seguirá el estándar descrito por la Ley Orgánica de Protección de Datos (LOPD), ya que la aplicación guardará datos personales del usuario como son el nombre, los apellidos y la contraseña que introdujo durante su registro.

6.3.3.1. *Restricciones generales*

Aquí se identifican restricciones que afectan al sistema de información a desarrollar. Las próximas restricciones han sido impuestas por el cliente, el desarrollador, normas generales, etc.

- **Restricciones de interfaz de usuario.**
 - *Amigable con el cliente.*
 - *No debe tener ambigüedades que lleven a la confusión*
 - *Debe tener las funcionalidades que aparecen descritas en los requisitos de usuario.*
- **Restricciones de codificación.**
 - *El lenguaje a usar será Java.*
 - *El IDE a usar será Eclipse.*
 - *La versión de la JDK debe ser la 1.7*
 - *El servidor de aplicaciones donde se desplegará el trabajo será WildFly 8.2*
- **Restricciones de seguridad y responsabilidad.**
 - *Todo elemento que no sea de carácter público deberá permanecer inaccesible y oculto.*

- *Los datos personales de los usuarios sólo podrán ser consultados por ellos mismos.*
- *La contraseña del usuario deberá guardarse de forma cifrada en la base de datos.*
- *Los usuarios con acceso a archivos compartidos sólo podrán realizar las acciones que les han sido concedidas, en caso de poder compartir el caso sólo podrán asignar como máximo los permisos que ellos tienen sobre él.*
- **Restricciones de eficiencia**
 - *El tiempo medio de carga y descompresión de casos tras login debe ser inferior a 5 segundos.*
 - *El tiempo medio de cierre y eliminación de archivos temporales tras el cierre de sesión debe ser inferior a 5 segundos.*
 - *El tiempo medio de carga de una página común debe ser inferior a 2 segundos.*
- **Restricciones de compatibilidad**
 - *El sistema deberá dar soporte para inglés y castellano.*
 - *La aplicación debe ser accesible desde cualquier navegador web.*
- **Restricciones de casos.**
 - *La extensión de los casos de simulación debe ser ONGF, ZIP o RAR.*
 - *Los casos que superen el tamaño de 10 MiB no podrán ser subidos al servidor.*
 - *Sólo se podrá subir un máximo de 10 casos de manera simultánea.*
 - *Todo caso deberá poder ser validado frente a su esquema XSD[6].*
- **Restricciones de base de datos**
 - *El gestor a usar será MySQL.*
 - *Deberán crearse únicamente las tablas que se especifiquen en el diseño.*
 - *La contraseña de los usuarios deberá guardarse cifrada.*

6.3.3.2. *Supuestos y dependencias*

En términos informáticos la aplicación web depende del servidor de simulaciones. Se deberán implementar mecanismos para controlar la disponibilidad de este y funcionalidades de comunicación para desarrollar el proceso de simulación. De la misma manera, si el servidor de simulaciones esta caído o inaccesible, la aplicación web deberá trabajar con normalidad y únicamente deshabilitará las funcionalidades que implique una actuación del servidor ajeno.

Externamente existe una dependencia con la Ley Orgánica de Protección de Datos, de modo que si esta es modificada deberá revisarse el proyecto por si se deben realizar modificaciones para adaptarlo.

6.3.3.3. *Entorno operacional*

El entorno tecnológico para el desarrollo es el identificado en el EVS como 4.4.1. *Alternativa 1: Java Servlets + MySQL*. El cual se desarrollará mediante el IDE de Eclipse Luna. El desarrollador del sistema, a parte de las herramientas anteriores, deberá usar el servidor de aplicaciones WildFly para desplegar la aplicación web, mientras que los usuarios únicamente necesitarán de

un navegador web (preferiblemente Google Chrome) de cualquier sistema operativo para acceder a ella.

Para el desarrollo del sistema se usará un equipo con las siguientes características:

- Procesador: Intel Core i5-3230M 2,6 GHz
- RAM: 6 GB
- Sistema operativo: Windows 8.1 64 bits

6.3.4. Identificación de los usuarios participantes y finales

Este apartado identifica a los usuarios participantes en el análisis del sistema, usuarios que validan y aceptan la aplicación y usuarios finales o stakeholders.

Los usuarios identificados para este proyecto han sido los siguientes:

- **Equipo de desarrollo:** formado únicamente por el autor de este TFG, Javier García Pérez, se encargará de programar el sistema y comprobar su correcto funcionamiento. También actuará como Project manager.
- **Usuarios finales:** cualquier persona que usará la aplicación web una vez finalizada.
- **Personal docente:** durante la elaboración del TFG y una vez finalizado el mismo tendrán acceso tanto el tutor del alumno, Israel González Carrasco, que tendrá que supervisar el correcto desarrollo del mismo como el resto del personal docente de la universidad que tenga que evaluarlo.

6.3.5. Estudio de la seguridad requerida en el proceso de análisis del sistema de información

Todos los productos y documentos generados durante las diferentes etapas de este TFG serán accesibles por las personas identificadas en el punto 6.3.4. *Identificación de los usuarios participantes y finales*.

Además, como ya ha hecho hincapié en algunos puntos del documento, toda la documentación desarrollada seguirá los estándares y normas de la metodología de desarrollo Métrica v3.

6.4. Establecimiento de requisitos software

Este punto tiene una gran importancia ya que es en el que se definen los requisitos software en base a los requisitos de usuarios planteados en el EVS. Todo requisito software debe tener una correspondencia con al menos un requisito de usuario, además, la definición de estos nuevos requisitos debe estar implementada de manera ordenada y correcta en la aplicación web.

Además, en este punto también se especificarán los casos de uso que se han desprendido de los requisitos de usuario.

6.4.1. Especificación de casos de uso

Un caso de uso proporciona de una manera clara y concisa la forma en la que debe interactuar con la aplicación algún actor del sistema para llevar a cabo un proceso descrito en los requisitos de usuario.

Todo caso de uso especificado en este documento vendrá acompañado de su respectivo diagrama y tabla, de esta manera se pretende conseguir la mayor claridad posible.

Un ejemplo de los diagramas y tablas que se usarán en adelante en este apartado son los siguientes:

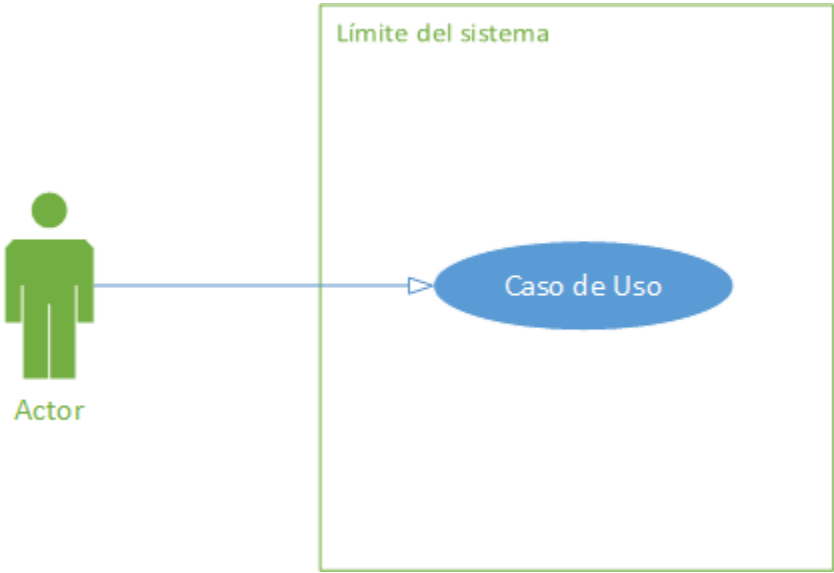


Ilustración 24 Diagrama ejemplo caso de uso

CU-XX	
Nombre	Nombre del caso
Actores	Usuarios que pueden realizar esa acción
Objetivo	Finalidad de la acción
Precondiciones	Condiciones que debe cumplir el actor para realizar la operación
Postcondiciones	Estado del sistema tras realizar la operación
Caso de uso general	
Pasos a seguir para realizar la tarea	
Casos Alternativos	
Circunstancias que pueden darse si no se realiza los pasos generales	

Tabla 50 Tabla ejemplo caso de uso

A continuación se describe con más detalle el significado de cada campo de la tabla mostrada anteriormente:

- Identificador:** Identifica unívocamente cada caso de uso. La norma a seguir es *CU-XX*, donde:

- *CU*: Caso de uso
- *XX*: numeración entera que irá incrementándose en una unidad.
- **Nombre:** Nombre identificativo del caso de uso.
- **Actores:** Usuarios que pueden interactuar con la aplicación para realizar la acción descrita.
- **Objetivo:** Finalidad de la operación que realiza el actor en ese caso.
- **Precondición:** Reglas y pautas que debe cumplir el actor para realizar la operación en el sistema.
- **Postcondición:** Reglas y pautas que deben cumplir el sistema y actor tras realizar la operación.
- **Caso de uso general:** Descripción por pasos de las tareas a realizar por el actor.
- **Casos alternativos:** Circunstancias y caminos que pueden darse si el actor no realiza un paso correctamente.

A continuación se especifican los diferentes casos de uso extraídos en base a los requisitos de usuario del EVS.

6.4.1.1. *CU-00: Acceso a la aplicación web*

Este caso muestra las acciones que los usuarios pueden realizar para acceder a la aplicación.

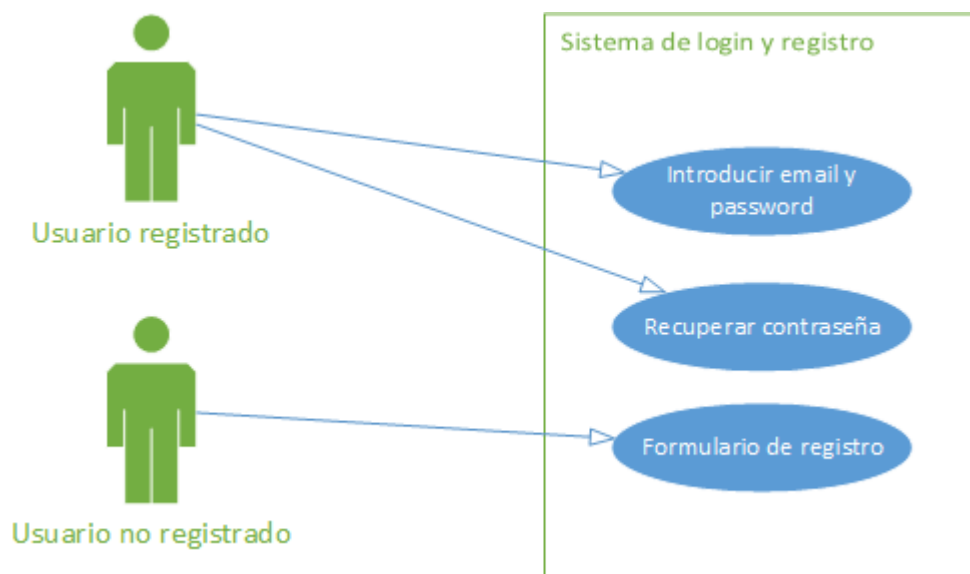


Ilustración 25 CU-00

CU-00	
Nombre	Acceso a la aplicación web
Actores	Usuario registrado y usuario no registrado
Objetivo	Acceder al repositorio del usuario en la aplicación
Precondiciones	El usuario debe haberse registrado
Postcondiciones	El usuario accede a su repositorio
Caso de uso general	
1-	El usuario introduce su email y contraseña

- 2- El sistema comprueba que los datos son correctos y carga los casos de simulación que tenga almacenados el usuario
- 3- El sistema da acceso al usuario a la aplicación web

Casos Alternativos

- 1- El usuario introduce su email y/o contraseña incorrectamente
 - 1.1- El sistema no da el visto bueno a los datos introducidos
 - 1.2- El sistema notifica al usuario que uno de los datos introducidos fue erróneo.
- 2- El usuario no recuerda su contraseña
 - 2.1- El usuario selecciona la opción de recuperar contraseña
 - 2.2- El sistema manda un correo al email introducido por el usuario con los pasos a seguir
- 3- El usuario no está registrado
 - 3.1- El usuario selecciona la opción de registrarse
 - 3.2- El sistema manda un correo al email de registro del usuario con la activación de su cuenta

Tabla 51 CU-00

6.4.1.2. CU-01: Cierre de sesión

Este caso muestra las acciones que los usuarios deben realizar para cerrar correctamente su sesión.

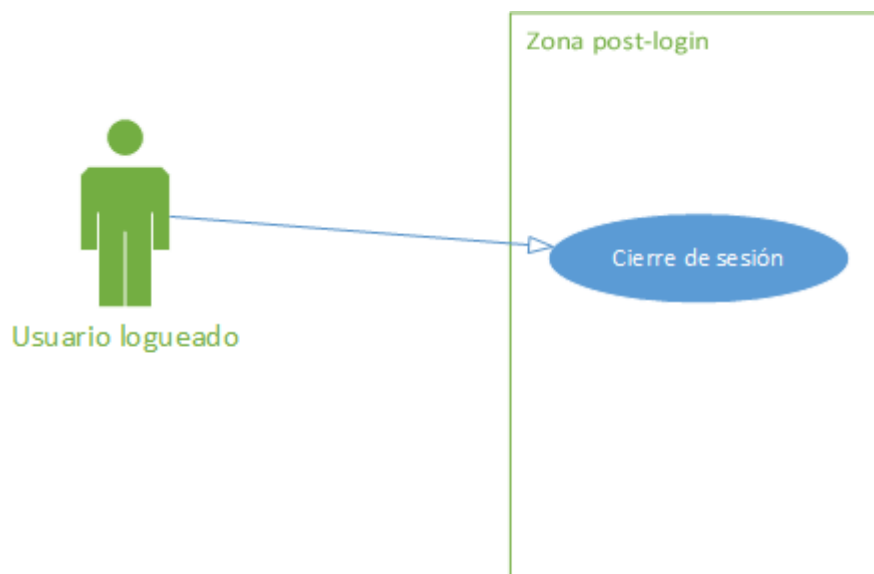


Ilustración 26 CU-01

CU-01	
Nombre	Cierre de sesión
Actores	Usuario logueado
Objetivo	Cerrar la sesión voluntariamente
Precondiciones	El usuario debe estar logueado
Postcondiciones	El usuario debe ser redirigido a la pantalla de login con su sesión cerrada
Caso de uso general	
1-	El usuario selecciona la opción de logout

- 2- El sistema borra todas las carpetas temporales asociadas a ese usuario e invalida la sesión
- 3- El sistema redirige al usuario a la pantalla de login

Casos Alternativos

- 1- El usuario cierra la ventana de la aplicación sin cerrar sesión o deja esta inactiva
 - 1.1- El sistema tras pasar 30 minutos la sesión inactiva la invalida
 - 1.2- Si el usuario intenta acceder tras ese tiempo a alguna pantalla que obligue a estar logueado, el sistema le redirigirá a la pantalla de login

Tabla 52 CU-01

6.4.1.3. CU-02: Gestión de cuenta

Este caso muestra las acciones que los usuarios pueden realizar para modificar alguno de sus datos de registro.

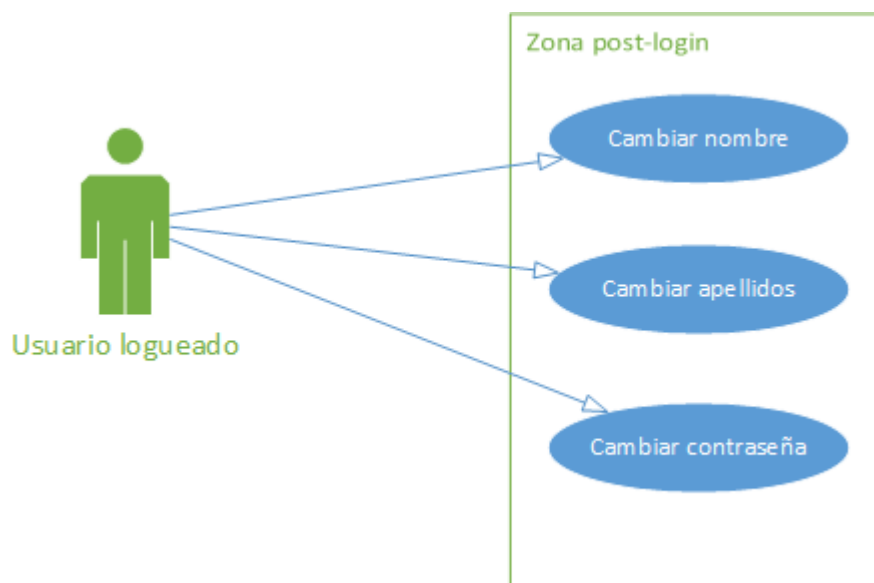


Ilustración 27 CU-02

CU-02	
Nombre	Gestión de cuenta
Actores	Usuario logueado
Objetivo	Cambiar algún dato de registro
Precondiciones	El usuario debe estar logueado
Postcondiciones	El usuario debe ser redirigido a la pantalla de gestión de cuenta con los cambios aplicados
Caso de uso general	
<ol style="list-style-type: none"> 1- El usuario selecciona la opción de <i>cuenta</i> 2- El sistema muestra el valor de los datos actuales 3- El usuario modifica los datos que desee entre el nombre, apellidos y contraseña 4- El sistema actualiza en la base de datos estos cambios y redirige al usuario a la pantalla de gestión de cuenta con estos aplicados 	
Casos Alternativos	

1-	El usuario intenta guardar los cambios con algún campo en blanco
1.1-	El sistema avisa al usuario de este hecho con un pop-up encima del campo vacío
1.2-	El sistema no realiza ninguna actualización de los datos
2-	El usuario cambia mal la contraseña
2.1-	El usuario introduce una nueva contraseña
2.2-	Los textos de los campos 'Contraseña' y 'Repetir contraseña' no coinciden
2.3-	El sistema deshabilita la opción de actualizar los datos

Tabla 53 CU-02

6.4.1.4. CU-03: Subir caso de simulación

Este caso muestra las acciones que los usuarios deben realizar para subir a su repositorio casos de simulación.

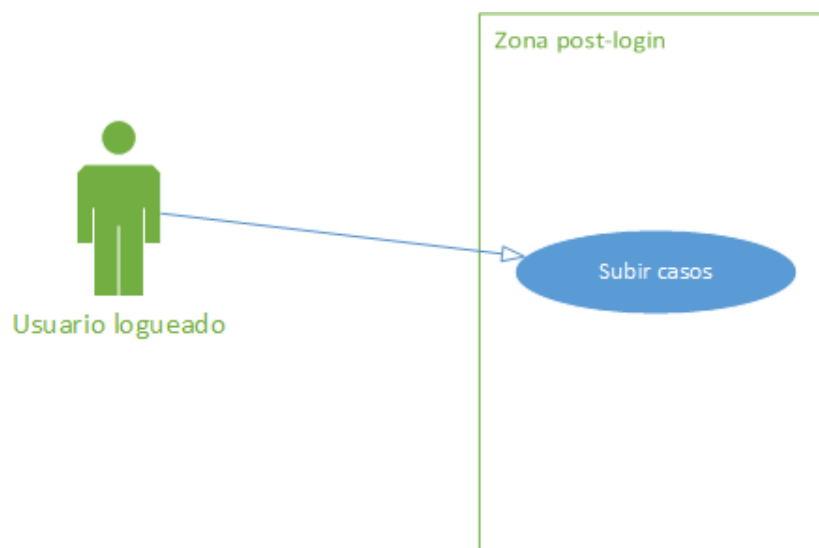


Ilustración 28 CU-03

CU-03	
Nombre	Subir caso de simulación
Actores	Usuario logueado
Objetivo	Alojar en el repositorio del usuario casos de simulación
Precondiciones	El usuario debe estar logueado y tener al menos un caso de simulación local
Postcondiciones	El usuario, tras la subida, debe ser redirigido a la pantalla principal del repositorio con los casos subidos ya apareciendo. Estos nuevos casos deben comenzar su simulación
Caso de uso general	
1-	El usuario selecciona la opción de subir caso
2-	El sistema muestra una ventana donde poder arrastrarlos y un botón para subirlos de manera tradicional[11]
3-	El usuario selecciona tantos casos como quiera subir (máximo 10 por subida)
4-	El sistema empezará a subir todos los casos y los añadirá a la cola de simulación.
5-	El sistema redirigirá a la pantalla del repositorio al usuario tras subir completamente todos los casos
Casos Alternativos	

1-	El usuario ya tiene un caso con el mismo nombre en el repositorio
1.1-	El sistema avisa al usuario de este hecho con un pop-up dándole a elegir si quiere reemplazar el caso más viejo o mantener los dos.
1.2-	El sistema redirige a la pantalla del repositorio tras la elección del usuario y tras haberse completado la subida.
2-	El usuario intenta subir un archivo inválido
2.1-	El usuario incluye en la carga de archivos un fichero que no cumple las normas
2.2-	El sistema no sube ese archivo y muestra un mensaje explicando el porqué.
3-	El servidor de simulaciones está caído o inaccesible
3.1-	El sistema no permite al usuario subir nuevos casos

Tabla 54 CU-03

6.4.1.5. CU-04: Gestión de casos

Este caso muestra las acciones que los usuarios pueden realizar para gestionar los casos almacenados en sus repositorios.

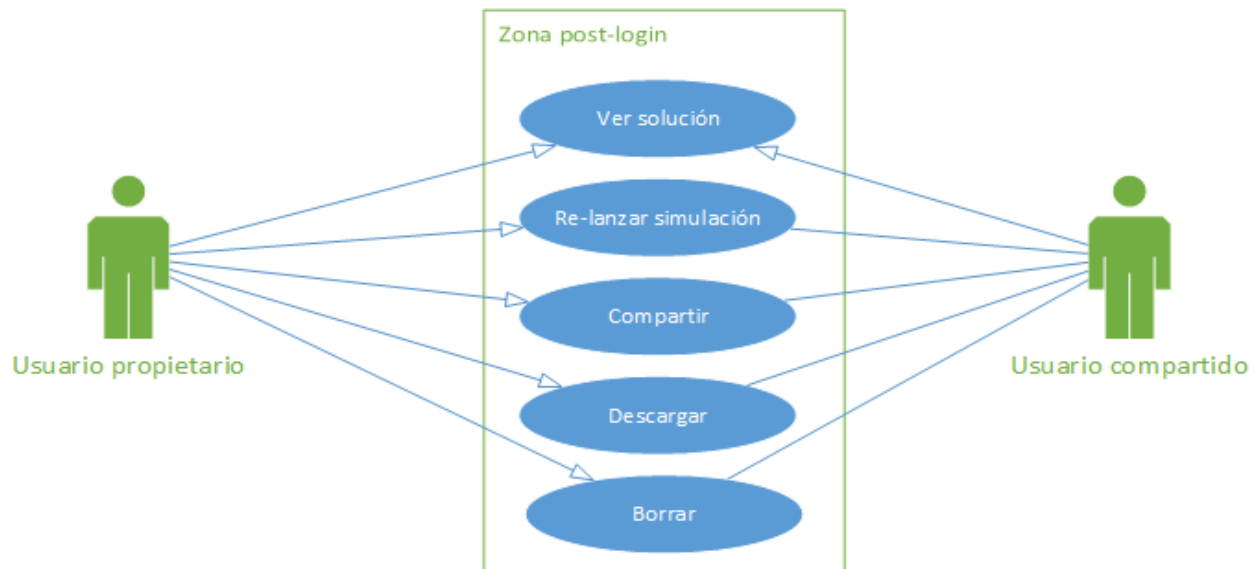


Ilustración 29 CU-04

CU-04	
Nombre	Gestión de casos
Actores	Usuario propietario y usuario compartido
Objetivo	Realizar diferentes funcionalidades sobre los casos de simulación subidos
Precondiciones	El usuario debe estar logueado, tener al menos un caso de simulación subido y ser propietario de él o tener acceso por compartición
Postcondiciones	El usuario debe poder ver el resultado de usar cada una de las funcionalidades
Caso de uso general	
1-	El usuario elige un caso de su repositorio
2-	El usuario elige una de las opciones de gestión (ver solución, re-lanzar simulación, compartir, descargar o borrar)
3-	El sistema realiza la funcionalidad relacionada con la opción elegida por el usuario
4-	El usuario ve el resultado de la opción elegida

Casos Alternativos	
1-	El usuario es compartido e intenta realizar una acción sin permisos
1.1-	El usuario intenta realizar una opción la cual el propietario no le dio permisos
1.2-	El sistema tiene deshabilitadas las opciones en las que el usuario no tiene permisos
2-	El servidor de simulaciones está caído o inaccesible
2.1-	El sistema no permite al usuario relanzar la simulación del proyecto
2.2-	El sistema no permite al usuario compartido relanzar la simulación del proyecto

Tabla 55 CU-04

6.4.1.6. CU-05: Gestión de permisos de caso

Este caso muestra las acciones que los usuarios deben realizar para gestionar los permisos de sus casos.

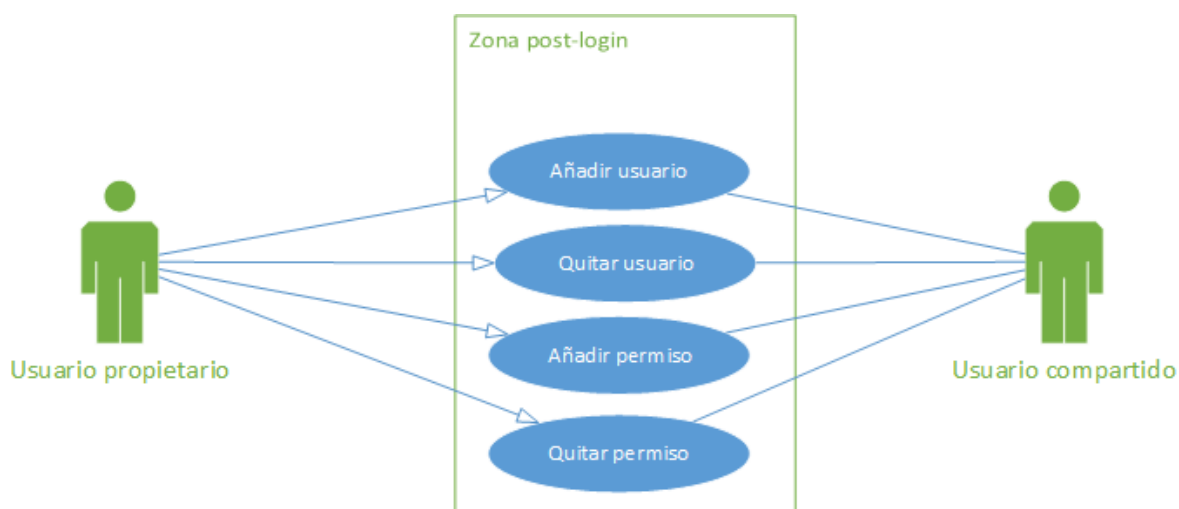


Ilustración 30 CU-05

CU-05	
Nombre	Gestión de permisos de caso
Actores	Usuario propietario y usuario compartido
Objetivo	Añadir/quitar usuarios y asignarles permisos sobre un caso
Precondiciones	El usuario debe estar logueado, tener al menos un caso de simulación subido y ser propietario de él o tener acceso por compartición
Postcondiciones	El usuario debe poder ver los permisos de cada usuario que él añadió sobre el caso
Caso de uso general	
1-	El usuario elige un caso de su repositorio
2-	El usuario elige la opción de compartir
3-	El sistema redirige al usuario a la pantalla de permisos de ese caso
4-	El usuario añade un nuevo usuario o gestiona los ya presentes
5-	El usuario elige que permisos dar y quitar y a que usuarios eliminar la compartición de ese archivo
Casos Alternativos	

1-	El usuario es compartido y asigna permisos a otro usuario
1.1-	El usuario compartido añade un nuevo usuario
1.2-	El sistema comprueba que este no es el propietario o el que le dio la compartición a este usuario
1.3-	El sistema solo habilita los mismos permisos a asignar a los que tiene acceso este usuario
2-	El dueño o superior de un caso revoca la compartición a otro usuario
2.1-	El sistema revoca en cascada los permisos que el usuario eliminado asigno a otros usuarios
2.2-	El usuario revocado y todos los que obtuvieron la compartición gracias a él ya no tienen acceso a ese caso

Tabla 56 CU-05

6.4.1.7. CU-06: Elegir idioma

Este caso muestra las acciones que los usuarios deben realizar para cambiar el idioma de la aplicación web.

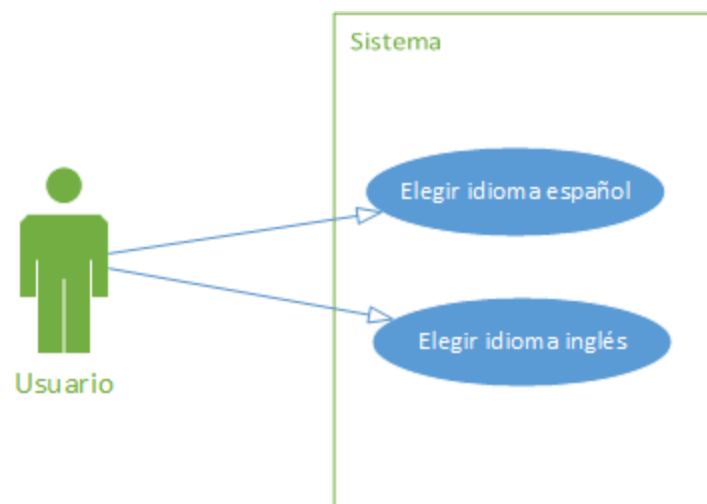


Ilustración 31 CU-06

CU-06	
Nombre	Elegir idioma
Actores	Usuario
Objetivo	Cambiar el idioma de la aplicación web
Precondiciones	El usuario debe navegar por la aplicación web
Postcondiciones	El sistema debe redirigir a la misma página en la que estaba el usuario tras aplicar el cambio de idioma
Caso de uso general	
1-	El usuario elige el idioma deseado
2-	El sistema redirige al usuario a la misma página con el idioma cambiado
3-	El usuario aprecia el cambio en los textos de la aplicación web
Casos Alternativos	
1-	Una página de la aplicación no tiene compatibilidad con otros idioma
1.1-	El sistema mostrará el contenido de esta página en el idioma por defecto (español)

Tabla 57 CU-06

6.4.1.8. CU-07: Navegación entre casos de simulación

Este caso muestra las acciones que los usuarios deben realizar para navegar entre todos los casos que se muestren en sus repositorios.

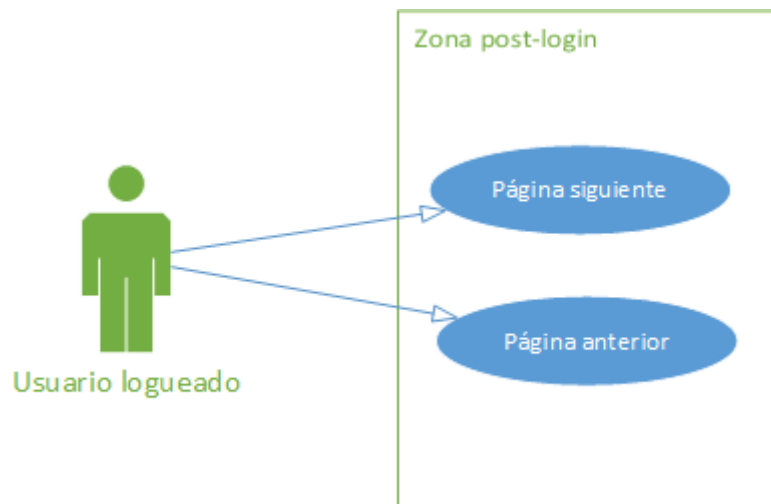


Ilustración 32 CU-07

CU-07	
Nombre	Navegación entre casos de simulación
Actores	Usuario logueado
Objetivo	Navegar entre todos los casos del repositorio
Precondiciones	El usuario estar logueado y tener más de cuatro casos propios o compartidos
Postcondiciones	El sistema debe mostrar la siguiente/anterior página de casos de simulación de ese usuario
Caso de uso general	
1- El usuario selecciona la siguiente/anterior página	
2- El sistema muestra la nueva vista con los casos de la siguiente/anterior página	
Casos Alternativos	
1- El usuario está en la primera	
1.1- El sistema solo mostrará la opción de página siguiente	
2- El usuario está en la última página	
2.1- El sistema sólo mostrará la opción de página anterior	

Tabla 58 CU-07

6.4.2. Obtención de requisitos

Este punto registra los requisitos software que se han extraído durante la fase de análisis, los cuales tomarán como base los requisitos de usuarios descritos en el EVS.

A continuación se explica la estructura y terminología que seguirán los requisitos software recogidos en este documento.

- **Requisito software (SR):** requisito que el software debe cumplir para garantizar su correcto funcionamiento, dentro de ellos hay diferentes tipos, los que usaremos en nuestro proyecto son los siguientes.
 - *Requisitos funcionales (F):* Definen el propósito del software y se derivan de los casos de uso, que están derivado de los requisitos de capacidad del usuario.
 - *Requisitos de calidad (C):* especifican los atributos del software que aseguran que será adecuado para su propósito.
 - *Requisitos de documentación (D):* especifican los requisitos específicos del proyecto para la documentación, además de los contenidos en los estándares.
 - *Requisitos de seguridad (S):* especifican los requisitos para asegurar el sistema contra amenazas de confidencialidad, la integridad y la disponibilidad.
 - *Requisitos de comprobación (M):* estos requisitos especifican las limitaciones que afectan a cómo el software debe verificar los datos de entrada y salida.
 - *Requisitos de rendimiento (R):* especifican valores numéricos para variables de rendimiento, como por ejemplo tasas de transferencia, frecuencia, capacidad y velocidad de proceso.
 - *Requisitos de interfaz (I):* especifican hardware y/o software con el que el sistema o componentes del sistema deben interactuar o comunicarse
 - *Requisitos de operación (O):* son aquellos que van a indicar cómo va a realizar el sistema las tareas para las que ha sido construido, garantizando los niveles de servicio requeridos.

Dada la clasificación anterior, los requisitos que se mostrarán a continuación llevarán los siguientes atributos asociados en sus tablas.

- **Identificador:** cada requisito será identificado unívocamente por un identificador del tipo RS-YYY, donde:
 - SR: requisito software
 - X: identificador del tipo de requisito. Se seguirá la misma nomenclatura que en el listado de tipos de arriba.
 - YY: número entero del requisito, se irá incrementando en una unidad por cada nuevo requisito de ese tipo.
- **Título:** nombre del requisito.
- **Fuente:** Indica que parte de los interesados ha identificado el requisito, si ha sido el cliente o el Project manager.
- **Procedencia:** referencia al requisito o requisitos de usuario de los que parten los requisitos software.
- **Descripción:** breve descripción del cometido del requisito.
- **Necesidad:** necesidad de cumplir el requisito. Tomará los siguientes valores esencial, deseable y opcional, los requisitos con necesidad esencial no serán negociables.
- **Prioridad:** medida de importancia para que el desarrollador pueda decidir la planificación a seguir. Tomará los valores de alta, media y baja.
- **Estabilidad:** algunos requisitos se pueden saber fijos sobre la vida esperada del software, mientras que otros pueden depender de las decisiones de diseño o implementación que se tomen durante el desarrollo.

- **Verificabilidad:** cada requisito software será verificable. Los valores que puede tomar son: alta, media y baja.

La tabla que servirá como plantilla para recoger los requisitos software de la aplicación web es la siguiente.

SR-XYX			
Título	Ejemplo de requisito	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	El sistema debe...		
Procedencia	UR-XYX		

Tabla 59 Plantilla requisitos software

6.4.2.1. Requisitos funcionales

A continuación se recogen los requisitos funcionales de la aplicación web.

SR-F00			
Título	Acceso	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	El acceso al contenido de la aplicación web estará restringido mediante login desde una GUI en la que un formulario pedirá el email y contraseña de registro.		
Procedencia	UR-C00, UR-C06		

Tabla 60 SR-F00

SR-F01			
Título	Registro	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	El registro en la aplicación será mediante un formulario cuyos campos serán: <ul style="list-style-type: none"> • Email • Nombre • Apellidos • Contraseña 		
Procedencia	UR-C02		

Tabla 61 SR-F01

SR-F02			
Título	Recuperar contraseña	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager

Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	La aplicación permitirá recuperar la contraseña introduciendo el email de registro en un formulario. Posteriormente habrá que seguir los pasos indicados en el correo que llegará al email introducido.		
Procedencia	UR-C01		

Tabla 62 SR-F02

SR-F03			
Título	Activación cuenta	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Tras un registro el sistema mandará un correo al usuario con un link para activar la cuenta. El sistema guardará la fecha de activación de la cuenta.		
Procedencia	UR-C03		

Tabla 63 SR-F03

SR-F04			
Título	Gestión de datos	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Los propios usuarios podrán modificar sus datos de registro (excepto el email) desde una GUI de la aplicación web.		
Procedencia	UR-C05, UR-I02		

Tabla 64 SR-F04

SR-F05			
Título	Repositorio	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	El repositorio de usuario que aparecerá tras el login mostrará los casos propios subidos y los compartidos. En el caso de estar vacío mostrará un mensaje.		
Procedencia	UR-C04		

Tabla 65 SR-F05

SR-F06			
Título	Login	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Cada vez que se realice un login satisfactoriamente el sistema descomprimirá el contenido HTML de los casos del usuario logueado en una carpeta temporal e irá habilitando la interacción con los casos del repositorio según vaya completando.		

Procedencia	UR-C00, UR-C04, UR-C06, UR-R11
--------------------	--------------------------------

Tabla 66 SR-F06

SR-F07			
Título	Logout	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Cada vez que se realice un logout satisfactoriamente el sistema borrará la carpeta temporal asociada a los usuarios que desconecten		
Procedencia	UR-R12		

Tabla 67 SR-F07

SR-F08			
Título	Gestión de casos	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	El sistema permitirá realizar todas las acciones que se puedan sobre los casos desde la propia aplicación web (simular, descargar, compartir, borrar, ver resultado)		
Procedencia	UR-C04, UR-C07, UR-C08, UR-C09, UR-C11, UR-C12, UR-C13, UR-C14		

Tabla 68 SR-F08

SR-F09			
Título	Presentación casos	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Los casos de simulación aparecerán en la interfaz de usuario mostrando su nombre, fecha de última modificación y una serie de iconos relacionados con las acciones que se pueden realizar sobre él.		
Procedencia	UR-C04, UR-C07, UR-C08		

Tabla 69 SR-F09

SR-F10			
Título	Contenido repositorio	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	La interfaz del repositorio del usuario mostrará a lo máximo cuatro casos propios y cuatro casos compartidos. En caso de exceder este número permitirá cambiar de página donde estarán el resto de los casos.		
Procedencia	UR-C04, UR-C07, UR-C08, UR-R18		

Tabla 70 SR-F11

SR-F11			
--------	--	--	--

Título	Gestión de permisos	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	El sistema proporcionará una interfaz en la que el usuario podrá gestionar los permisos de los casos de simulación que comparta		
Procedencia	UR-C09, UR-R08, UR-I00		

Tabla 71 SR-F11

SR-F12			
Título	Idioma	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	La aplicación debe permitir seleccionar entre el idioma español e inglés. El idioma por defecto será el español.		
Procedencia	UR-C05, UR-C16, UR-R10		

Tabla 72 SR-F12

SR-F13			
Título	Sesión varios ordenadores	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Cuando el usuario realice el login el sistema debe cerrar las sesiones que haya activas de ese usuario en otros ordenadores		
Procedencia	UR-C00, UR-C06		

Tabla 73 SR-F13

SR-F14			
Título	Menú aplicación	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	<p>El menú superior de la aplicación web deberá mostrar el selector del idioma, tras login este menú también mostrará:</p> <ul style="list-style-type: none"> • Estado del servidor de simulaciones • Nombre y apellidos del usuario • Enlace a gestión de cuenta • Enlace a cierre de sesión 		
Procedencia	UR-C05, UR-C16, UR-R01		

Tabla 74 SR-F14

SR-F15			
Título	Estado simulación	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager

Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	<p>Durante la simulación de un caso el sistema debe mostrar el estado de esta a los usuarios con acceso a él. Los posibles estados son:</p> <ul style="list-style-type: none"> • Pendiente: el servidor de simulaciones recibió el caso pero todavía no lanzó la simulación. • Calculando: Realizando simulación. • Simulado: Simulación simulado. El usuario ya tendría acceso a las funcionalidades del caso. • Errores: debido a un fallo durante el cálculo o a un erro durante la comunicación con el servidor. 		
Procedencia	UR-C07, UR-C10, UR-C12, UR-C18		

Tabla 75 SR-F15

SR-F16			
Título	Subida de casos	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	El sistema proporcionará una interfaz al usuario en la que podrá subir casos al servidor por medio de la técnica drag&drop o por la manera tradicional.		
Procedencia	UR-C10, UR-R03, UR-R06, UR-R07		

Tabla 76 SR-F16

SR-F17			
Título	Comprobación casos	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	El sistema comprobará que un caso cumple con los estándares y normas especificadas en este documento antes de alojarlo en el servidor.		
Procedencia	UR-10, UR-C17, UR-R03, UR-R07, UR-R09		

Tabla 77 SR-F17

SR-F18			
Título	Duplicidad de casos	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	<p>Si un caso cumple los estándares y normas el sistema comprobará antes e alojarlo definitivamente en el servidor si este caso ya existe con el mismo nombre en el repositorio del usuario. Si se da el caso entonces se dará a elegir al propio usuario entre:</p> <ul style="list-style-type: none"> • Reemplazar versión antigua y conservar sólo la nueva. • Mantener las dos versiones, el nombre de la nueva tendrá al final de este un número que indicará el número de la copia. 		
Procedencia	UR-10, UR-C17		

Tabla 78 SR-F18

SR-F19			
Título	Subida satisfactoria	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Si un caso se sube al servidor correctamente el sistema inmediatamente lo enviará al servidor de simulaciones para lanzar la simulación de este.		
Procedencia	UR-C10		

Tabla 79 SR-19

SR-F20			
Título	Borrado de casos	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	El sistema siempre pedirá confirmación cuando un usuario vaya a borrar un caso		
Procedencia	UR-C07, UR-C14, UR-C15		

Tabla 80 SR-F20

SR-F21			
Título	Resultado casos	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	El sistema mostrará en una pantalla diferente el resultado de un caso simulado		
Procedencia	UR-C07, UR-C11		

Tabla 81 SR-F21

6.4.2.2. Requisitos de calidad

A continuación se recogen los requisitos de calidad del proyecto.

SR-C00			
Título	Estabilidad de la aplicación	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Se debe garantizar una estabilidad y disponibilidad del sistema la cual debe ser 99,9 %, es decir, el tiempo máximo de caída del servicio no puede exceder de 43.8 minutos al mes u 8,76 horas al año.		
Procedencia	-		

Tabla 82 SR-C00

SR-C01			
Título	Copias de seguridad	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Se usará el repositorio GIT de jazzhub para controlar las versiones de la aplicación web		
Procedencia	-		

Tabla 83 SR-C01

SR-C02			
Título	Requisitos	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Todos los requisitos serán verificados de forma que se cumpla su propósito		
Procedencia	-		

Tabla 84 SR-C02

6.4.2.3. Requisitos de documentación

A continuación se recogen los requisitos de documentación del proyecto.

SR-D00			
Título	Manual de usuario	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	El proyecto incluirá un manual de usuario en castellano con la descripción detallada de las principales funcionalidades del sistema.		
Procedencia	-		

Tabla 85 SR-D00

SR-D01			
Título	Idioma documentación	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	La documentación desarrollada durante la realización del TFG estará en castellano		
Procedencia	-		

Tabla 86 SR-D01

6.4.2.4. Requisitos de seguridad

A continuación se recogen los requisitos de seguridad de la aplicación web.

SR-S00			
--------	--	--	--

Título	Campos de contraseñas	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	La aplicación mostrará el contenido de los campos de contraseñas oculto mediante *		
Procedencia	UR-C00, UR-C03, UR-I03		

Tabla 87 SR-S00

SR-S01			
Título	Contraseña en BD	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	La aplicación no guardará la contraseña explícita en la base de datos, sino la cifrará con MD5		
Procedencia	UR-I03, UR-R16		

Tabla 88 SR-S01

SR-S02			
Título	IDs usuarios y casos	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Por cada usuario registrado y caso subido se guardarán sus metadatos en la base datos y a los cuales se les asignará una ID única dentro de su grupo, con el fin de tenerlos identificados unívocamente.		
Procedencia	UR-R16		

Tabla 89 SR-S02

SR-S03			
Título	Permisos de casos	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Los usuarios con acceso compartido a casos sólo podrán realizar las acciones que les asigne el usuario que se lo compartió		
Procedencia	UR-R04, UR-R05, UR-I00		

Tabla 90 SR-R03

6.4.2.5. Requisitos de comprobación

A continuación se recogen los requisitos de comprobación de la aplicación web.

SR-M00			
Título	Comprobar login	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Para validar el proceso de login de un usuario el sistema a de comprobar que los siguientes datos coinciden con los guardados de un usuario: <ul style="list-style-type: none"> Email de entrada existente MD5 de la contraseña entrante coincidente con la almacenada y asociada al email introducido 		
Procedencia	UR-R11		

Tabla 91 SR-M00

SR-M01			
Título	Usuario compartir	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Cuando un usuario vaya a compartir un proyecto el sistema verificará que el email introducido se corresponde con un usuario existente y distinto al usuario que quiere compartir. En caso incorrecto se notificará el fallo.		
Procedencia	UR-C09		

Tabla 92 SR-M01

SR-M02			
Título	Email recuperar	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Cuando usuario realice la acción de recuperar contraseña el sistema comprobará que email existe en los registros antes de mandar el mensaje. En caso de no existir se notificará		
Procedencia	UR-C01		

Tabla 93 SR-M02

SR-M03			
Título	Sesión activa	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Si un usuario cierra la aplicación y al rato quiera acceder directamente a una página que requiera login el sistema verificará si hay una sesión activa, si existe le redirigirá a la página deseada, en caso contrario se le informará que debe loguear		
Procedencia	UR-C00		

Tabla 94 SR-M03

SR-M04			
Título	Formato casos	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional

Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	El sistema sólo aceptará la subida de casos de simulación cuyo formato de archivo sea <i>ONGF</i> , <i>ZIP</i> o <i>RAR</i> y cuyo tamaño no exceda los 10 MB. Además deberá ser validado contra un esquema XSD		
Procedencia	UR-R03, UR-R07, UR-R09		

Tabla 95 SR-M04

SR-M05			
Título	Iconos enabled	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	<p>Los tipos de iconos para interactuar con un caso que estarán activos son:</p> <ul style="list-style-type: none"> • Todos si el usuario es propietario del caso y este ha sido ya simulado • El de eliminar si el usuario es propietario del caso y este tuvo errores en su simulación. • Todos menos simular si el usuario es propietario del caso, ya ha sido simulado y el servidor de simulaciones está caído o inaccesible. • Todos si el usuario es compartido, el caso ha sido simulado y tiene todos los permisos sobre el caso. • Los correspondientes a los permisos asignados si el usuario es compartido y el caso ha sido simulado <p>En las circunstancias que no se recogen se da por hecho que permanecerán desactivados.</p>		
Procedencia	UR-R04, UR-R05, UR-R08		

Tabla 96 SR-M05

SR-M06			
Título	Log de simulación	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Por cada simulación lanzada de un caso el sistema debe guardar la traza generada en un archivo log que se guardará en el servidor con el nombre de la ID del caso		
Procedencia	UR-C12, UR-C18		

Tabla 97 SR-M06

6.4.2.6. Requisitos de rendimiento

A continuación se recogen los requisitos software del sistema.

SR-R00			
Título	Carga tras login	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema

Descripción	El tiempo medio de carga de la página de login debe ser inferior a 5 segundos (se incluye en este tiempo el proceso de descompresión de los casos que tenga ese usuario almacenados en el servidor)
Procedencia	UR-R11, UR-R17

Tabla 98 SR-R00

SR-R01			
Título	Carga tras logout	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	El tiempo medio de cierre y eliminación de archivos temporales tras el cierre de sesión debe ser inferior a 5 segundos.		
Procedencia	UR-R12, UR-R17		

Tabla 99 SR-R01

SR-R02			
Título	Carga página normal	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	El tiempo medio de carga de una página común debe ser inferior a 2 segundos.		
Procedencia	UR-R00, UR-R01, UR-R02, UR-I01		

Tabla 100 SR-R02

SR-R03			
Título	Fallo Heartbeat	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Si el Heartbeat lanzado hacía el servidor de simulaciones falla un total de 10 veces seguidas se mandará un correo al administrador web notificando el problema		
Procedencia	UR-R14		

Tabla 101 SR-R03

SR-R04			
Título	Fallo simulación	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Si al lanzar una simulación el servidor de simulaciones no recoge el proyecto y se han realizado más de 10 intentos se mandará un correo al administrador web notificando el problema		
Procedencia	UR-R13		

Tabla 102 SR-R04

6.4.2.7. Requisitos de interfaz

A continuación se recogen los requisitos de interfaz de la aplicación web.

SR-I00			
Título	Enviar correos	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	El sistema mandará correos electrónicos en los casos siguientes. <ul style="list-style-type: none"> • Recuperación de contraseña de usuario (al usuario). • Activación de cuenta de usuario (al usuario). • Aplicación web caída (al administrador web) • Servidor de simulaciones caído (al administrador web) • Simulación interrumpida por causa ajena (al administrador web) 		
Procedencia	UR-C01, UR-C03, UR-C18, UR-R14		

Tabla 103 SR-I00

SR-I01			
Título	Servidor simulaciones	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	La aplicación web se comunicará mediante WebServices con el servidor de simulaciones en los siguientes casos. <ul style="list-style-type: none"> • Simulación de un caso. • Proceso de Heartbeat. 		
Procedencia	UR-C18, UR-R13. UR-R14		

Tabla 104 SR-I01

SR-I02			
Título	Comunicación BD	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Todo almacenamiento y actualización de datos del usuario o casos de simulación se guardará en una base de datos que usara el gestor MySQL		
Procedencia	UR-R16		

Tabla 105 SR-I02

SR-I03			
Título	Peticiones cliente	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Toda petición de un usuario irá redirigida a un servlet específico que actuará como front-controller y redirigirá la petición donde proceda		

Procedencia	UR-R06
--------------------	--------

Tabla 106 SR-I03

6.4.2.8. Requisitos de operación

A continuación se recogen los requisitos de operación de la aplicación web.

SR-O00			
Título	Email recuperación	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	El email de recuperación de contraseña contendrá un texto explicativo y un link donde el usuario podrá resetear esta.		
Procedencia	UR-C01		

Tabla 107 SR-O00

SR-O01			
Título	Email activación	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	El email de activación de cuenta contendrá un texto explicativo y un link donde el usuario podrá activar esta.		
Procedencia	UR-C03		

Tabla 108 SR-O01

SR-O02			
Título	Request	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	El sistema recibirá las peticiones de los usuarios por medio de solicitudes que emplearán los métodos GET y POST		
Procedencia	UR-R06		

Tabla 109 SR-O02

SR-O03			
Título	Tratamiento request	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Toda solicitud que llegue al sistema será redirigida a un servlet que actuará como front-controller. En él se comprobará si el usuario que lanza la petición es correcto y posteriormente se redirigirá al servlet que debe tratar la petición en concreto.		

Procedencia	UR-R06
--------------------	--------

Tabla 110 SR-O03

SR-O04			
Título	Almacenamiento casos	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Los casos subidos por los usuarios se almacenarán en una carpeta en el servidor llamada <i>ONGFiles</i> . En ella cada usuario tendrá una carpeta fija cuyo nombre será la ID de este. Dentro de ella estarán los casos de los que es propietario que se guardarán en su estado comprimido y tendrán como nombre la ID asignada al caso. Cuando un usuario o el administrador borra un caso este desaparece de la carpeta de su propietario y pasa a la carpeta '0' dentro de <i>ONGFiles</i> que será la que almacene los casos borrados.		
Procedencia	UR-R15, UR-R17		

Tabla 111 SR-O04

SR-O05			
Título	Tratamiento casos	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	Cuando un usuario se loguea el sistema descomprimirá en una carpeta del servidor llamada <i>temp</i> todo el contenido HTML de los casos a los que este usuario tenga acceso, dentro de la carpeta <i>temp</i> habrá un directorio con la ID del usuario y dentro de él otro directorio con la ID de los casos que actualmente están descomprimidos. Tras el cierre de sesión se borrará la carpeta temporal asociada al usuario.		
Procedencia	UR-R17		

Tabla 112 SR-C05

SR-O06			
Título	Proceso Heartbeat	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema

Descripción	<p>El sistema hará el proceso de Heartbeat desde un planificador que mandará un mensaje al servidor de simulaciones mediante WebServices cada 2 segundos. Dicho mensaje estará compuesto de:</p> <ul style="list-style-type: none"> • Número de secuencia: número entero largo aleatorio. • Comando: 'IAMALIVE' en este caso. • Parámetros: 'FRONTEND' en este caso. <p>El servidor de simulaciones tras ver el comando y el parámetro del mensaje deberá responder a la aplicación con un mensaje que tendrá:</p> <ul style="list-style-type: none"> • Número de secuencia: mismo que el del mensaje recibido. • Parámetro: 'OK' <p>El sistema tras recibir la respuesta dará válido el Heartbeat. En el caso de no recibir una respuesta en el intervalo marcará el servidor de aplicaciones como caído o inaccesible y si la caída se prolongara durante los 10 próximos Heartbeat se notificará mediante email al administrador web del problema.</p>
	Procedencia UR-R13, UR-R14

Tabla 113 SR-006

SR-007			
Título	Inicio simulación	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema
Descripción	<p>El sistema hará el inicio de simulación de casos desde un planificador que se activará tras lanzar el propio usuario la simulación o tras subir un nuevo caso a la aplicación web. Este planificador mandará el caso al servidor, y parará una vez reciba confirmación de que le llegó la petición. Si pasan 5 segundos sin respuesta el sistema volverá a mandar la petición al servidor. Si no hay respuesta en 10 peticiones seguidas se mandará un email al administrador web.</p>		
Procedencia	UR-R13		

Tabla 114 SR-007

SR-008			
Título	Proceso de simulación	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Project manager
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Estabilidad	Durante toda la vida del sistema

Descripción	<p>Durante el proceso de simulación el sistema y el servidor de simulaciones intercambiarán mensajes mediante WebServices con la siguiente estructura. El primer caso será de peticiones y avisos, el siguiente de respuestas.</p> <ul style="list-style-type: none"> • Número de secuencia: número entero largo aleatorio. • Comando: 'LAUNCHCASE', 'CASEINQUEUE', 'CASELAUNCHED' o 'CASEFINISHED' • Parámetros: ID del caso <p>Respuesta:</p> <ul style="list-style-type: none"> • Numero de secuencia: mismo que la petición. • Parámetro: 'OK' <p>El significado de los comandos de las peticiones es el siguiente:</p> <ul style="list-style-type: none"> • LAUNCHCASE: petición del front-end para lanzar un caso. • CASEINQUEUE: aviso del servidor de que el caso está en cola. • CASELAUNCHED: aviso del servidor de que el caso se está simulando. • CASEFINISHED: aviso del servidor de que la simulación ha finalizado. <p>Los avisos de CASEINQUEUE y CASELAUNCHED no precisan de respuesta del front-end. Un ejemplo de la traza de la simulación sería el siguiente, a izquierda mensajes del front-end a la derecha del servidor:</p> <pre> 405624191 LAUNCHCASE 129008 405624191 OK 6675624117 CASEINQUEUE 129008 9765624722 CASEINQUEUE 129008 ... 88265624765 CASELAUNCHED 129008 75265896247 CASELAUNCHED 129008 ... 65624739776 CASEFINISHED 129008 65624739776 OK </pre> <p>El resultado de esta traza deberá guardarse en un log en el servidor con el nombre de la ID del caso.</p>
Procedencia	UR-R13

Tabla 115 SR-008

6.4.2.9. Matriz de trazabilidad: UR-SR

En la siguiente página se adjunta una matriz de trazabilidad para comprobar al detalle la relación de cada requisito software con su requisito de usuario.

	UR-C00	UR-C01	UR-C02	UR-C03	UR-C04	UR-C05	UR-C06	UR-C07	UR-C08	UR-C09	UR-C10	UR-C11	UR-C12	UR-C13	UR-C14	UR-C15	UR-C16	UR-C17	UR-C18	UR-R00	UR-R01	UR-R02	UR-R03	UR-R04	UR-R05	UR-R06	UR-R07	UR-R08	UR-R09	UR-R10	UR-R11	UR-R12	UR-R13	UR-R14	UR-R15	UR-R16	UR-R17	UR-R18	UR-I00	UR-I01	UR-I02	UR-I03	
SR-F00	X						X				X																																
SR-F01			X																																								
SR-F02		X																																									
SR-F03				X																																							
SR-F04					X																																				X		
SR-F05					X																																						
SR-F06	X				X		X																									X											
SR-F07																																		X									
SR-F08					X			X	X	X		X	X	X	X																												
SR-F09					X			X	X																																		
SR-F10					X			X	X																															X			
SR-F11										X																		X												X			
SR-F12						X											X													X													
SR-F13	X						X																																				
SR-F14						X											X				X																						
SR-F15								X			X		X						X																								
SR-F16											X												X			X	X																
SR-F17											X												X				X		X														
SR-F18											X							X																									
SR-F19											X																																
SR-F20								X							X	X																											
SR-F21							X					X																															

Tabla 116 Matriz trazabilidad UR-SR (1)

	UR-C00	UR-C01	UR-C02	UR-C03	UR-C04	UR-C05	UR-C06	UR-C07	UR-C08	UR-C09	UR-C10	UR-C11	UR-C12	UR-C13	UR-C14	UR-C15	UR-C16	UR-C17	UR-C18	UR-R00	UR-R01	UR-R02	UR-R03	UR-R04	UR-R05	UR-R06	UR-R07	UR-R08	UR-R09	UR-R10	UR-R11	UR-R12	UR-R13	UR-R14	UR-R15	UR-R16	UR-R17	UR-R18	UR-I00	UR-I01	UR-I02	UR-I03	
SR-S00	X	X																																									X
SR-S01																																				X							
SR-S02																																				X							
SR-S03																								X	X															X			
SR-M00																																X											
SR-M01										X																																	
SR-M02			X																																								
SR-M03	X																																										
SR-M04																							X				X		X														
SR-M05																								X	X			X															
SR-M06													X						X																								
SR-R00																															X								X				
SR-R01																																X						X					
SR-R02																				X	X	X																					
SR-R03																																			X						X		
SR-R04																																		X									
SR-I00			X		X														X															X									
SR-I01																			X														X	X									
SR-I02																																					X						
SR-I03																											X																
SR-O00			X																																								
SR-O01				X																																							
SR-O02																										X																	
SR-O03																										X																	
SR-O04																																				X			X				
SR-O05																																							X				
SR-O06																																		X	X								
SR-O07																																		X									
SR-O08																																		X									

Tabla 117 Matriz trazabilidad UR-SR (2)

6.5. Identificación de subsistema de análisis

En este proyecto el lenguaje base que se usará para el desarrollo de su funcionalidad es Java. El cual es orientado a objetos, estos lenguajes se caracterizan principalmente por dos cualidades: abstracción y reutilización. El concepto de clase como base para un objeto proporciona ventajas al combinar la abstracción funcional y la abstracción de datos. Así al contener tanto a los datos como a las funciones que comportan sobre los datos, se consigue reutilizar y mantener un elemento software.

Las clases se pueden asociar también en grupos de clases, permitiendo que se agrupen y se traten como un único elemento lógico proporcionando un modelo de reutilización efectivo a nivel de sistema, teniendo en cuenta el número de dependencias que existen entre ellos.

Tras esta breve introducción, se pretende aclarar en este punto el metapatrón elegido para la descomposición en subsistemas del proyecto. En la siguiente sección se identificara el metapatrón elegido con el fin de abordar el desarrollo de la aplicación de una manera más sencilla y organizada.

6.5.1. Determinación de subsistemas de análisis

La elección de los subsistemas se ha elegido en base a una separación funcional de las tareas del sistema.

Se seguirá un metapatrón (conjunto de patrones). El cual estará compuesto por el patrón de arquitectura MVC (Model-View-Controller) y el patrón Front-Controller. Antes de adentrarse en diagramas y descomposiciones se hará un breve resumen de cada uno.

El patrón de arquitectura MVC define la organización independiente del modelo (Objetos de negocio), la vista (interfaz de usuario) y el controlador (controlador del workflow de la aplicación). Adentrándose más en sus tres partes:

- **Modelo:**
 - Contiene el núcleo de la funcionalidad de la aplicación.
 - Encapsula el estado de la aplicación.
 - Independiente del controlador y la vista.
- **View:**
 - Es la presentación del modelo.
 - Puede acceder al modelo pero nunca cambiar su estado.
 - Puede ser notificada cuando hay un cambio de estado en el modelo.
- **Controlador:**
 - Reacciona a la petición del cliente, ejecutando la acción adecuada y creando el modelo pertinente.

Un ejemplo de la adaptación de este patrón a una aplicación Java Servlets como la tratada en esta memoria es la siguiente:

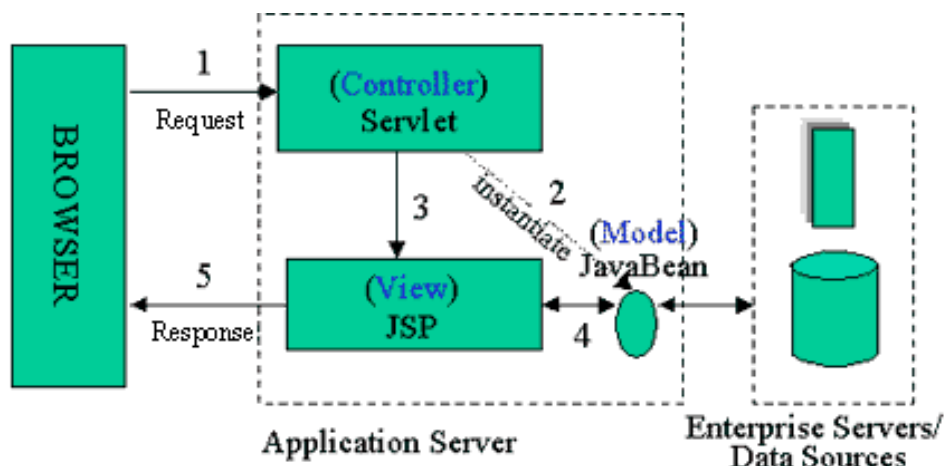


Ilustración 33 MVC - Java Servlets | Recuperado de: <http://www.javaworld.com/article/2076557/java-web-development/understanding-javascript-pages-model-2-architecture.html>

El patrón Front-Controller[9] recomienda el uso de un punto centralizado de acceso al sistema para el tratamiento de peticiones web (request). Esto es debido a que si no tuviera un acceso centralizado la navegación del sistema queda definida (en código) en las vistas o páginas JSPs, lo cual produce código duplicado y difícil de mantener debido a la mezcla de presentación con navegación.

Este patrón en el entorno de una aplicación web, supone tener un servlet controlador como único punto de entrada a la aplicación, este no produce salida del sistema, sólo redirige la solicitud a un subcontrolador. Este centraliza todas las peticiones entrantes de los clientes, remitiendo cada petición al gestor de peticiones o dispatcher adecuado que se encargará de gestionar la respuesta al cliente.

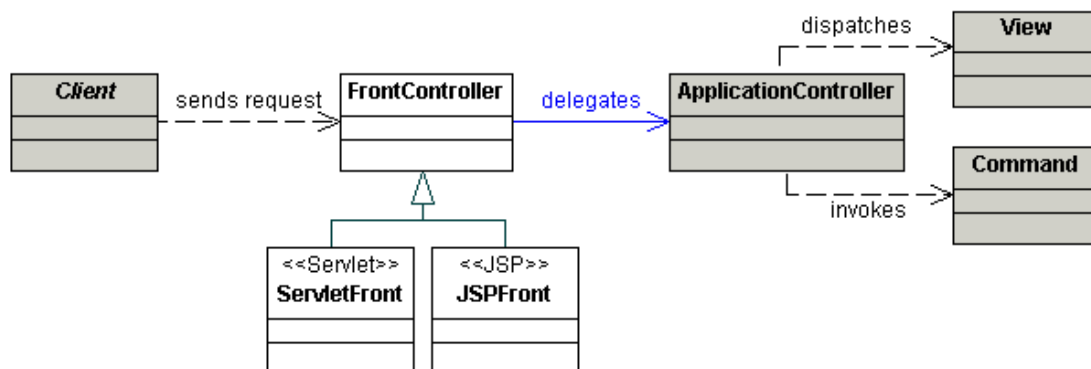


Ilustración 34 Front-Controller | Recuperado de: <http://corej2eepatterns.com/FrontController.htm>

La imagen anterior muestra su funcionamiento, una petición del cliente llega al front-controlador, este la evalúa y la redirige a un subcontrolador que debe procesarla.

A continuación se adjuntan una serie de diagramas con la adaptación de estos patrones a nuestro sistema.

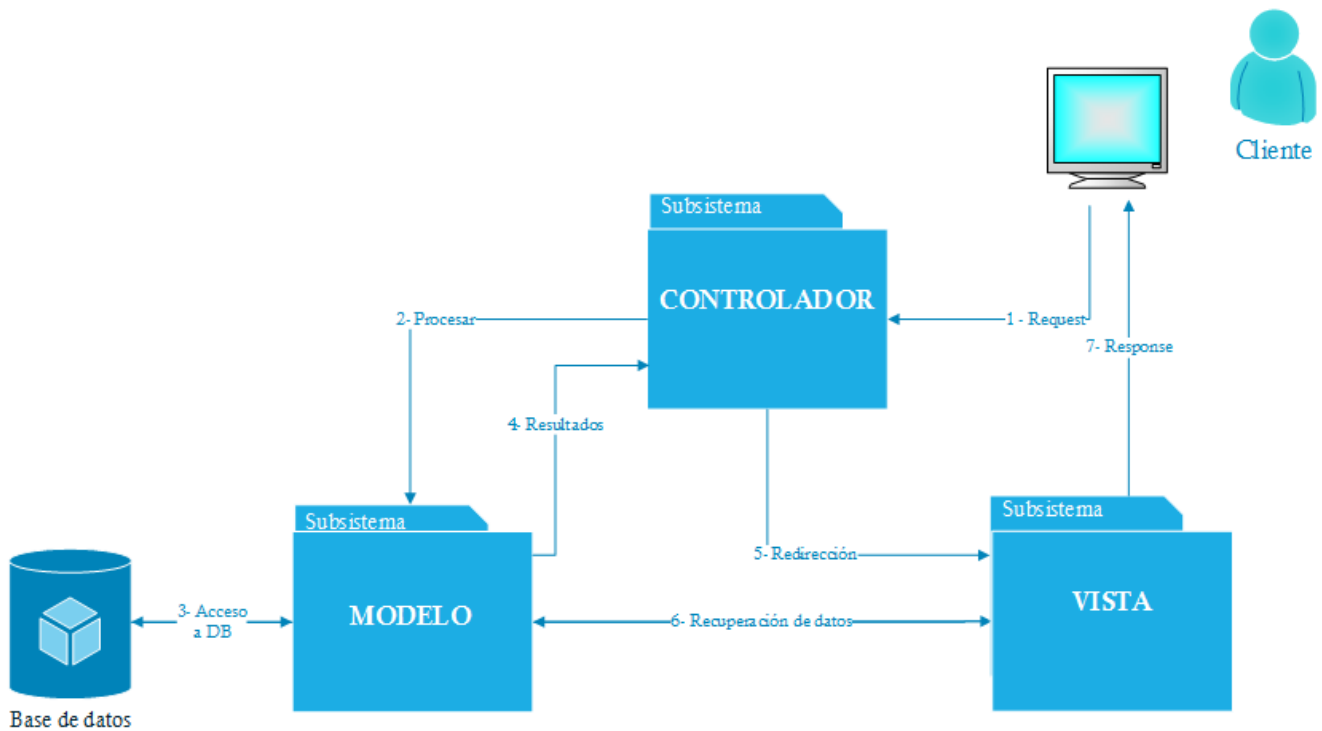


Ilustración 35 MVC-Sistema

- Subsistema controlador: Estará compuesto por las clases que atenderán las peticiones de los clientes.
- Subsistema modelo: Contendrá los objetos DAO para interactuar con la base de datos (se usará JPA) y almacenará la lógica de negocio.
- Subsistema vista: Formado por JSP que muestran el resultado final de la petición del cliente.

Dentro de cada subsistema se puede encontrar los siguientes componentes:

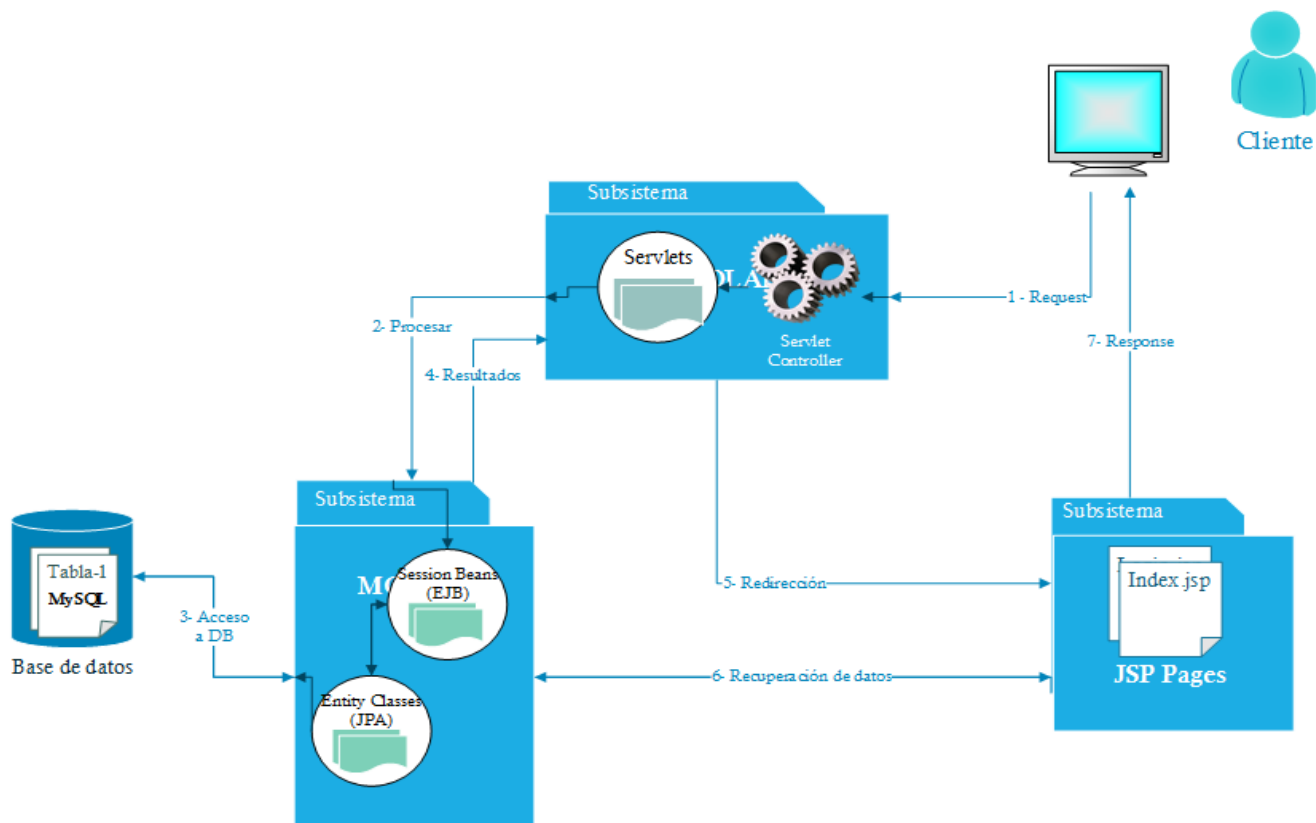


Ilustración 36 MVC-Sistema en detalle

En resumen, toda petición del cliente llegará al controlador y será tratada por el servlet-controller, el cual redirigirá la petición al servlet correspondiente, este procesará la petición mandándosela a su EJB, el cual, apoyándose en JPA para interactuar con la base de datos, logrará unos resultados que serán devueltos al controlador y este a su vez enviará a la vista donde el usuario podrá ver el resultado de su petición

Aclarar que este proyecto es una primera versión del sistema y como no comparte módulos con otros subsistemas para que se diera el caso de compartición y reutilización de EJBs se ha decidido que se implementará la lógica de negocio dentro de los servlets a los que redirigirá el servlet-controller, con el objetivo de reducir la complejidad. Se seguirá usando JPA para comunicarse con la base de datos.

6.6. Análisis de los casos de uso

En esta actividad se van a estudiar los casos de uso que se han definido en la actividad 6.4.1. *Especificación de casos de uso*, para identificar las clases cuyos objetos son necesarios para realizarlo y describir su comportamiento mediante la interacción de dichos objetos.

6.6.1. Identificación de clases asociadas a un caso de uso

En este punto se va a iniciar la identificación de los objetos necesarios para realizar cada caso de uso. Para ello se van a estudiar los casos de uso identificados anteriormente y de ellos se extraerá una lista de objetos candidatos a ser clases.

En esta primera extracción de objetos cabe la posibilidad de que no se obtengan todas las clases que posteriormente compondrán el software, pero esta fase servirá como aproximación, será después en la actividad del diseño donde se formalizará más el total de clases a usar.

En esta actividad se describirán tres tipos de clases que pueden extraerse de los casos de usuario, estas son:

- **Clases de entidad:** representan la información manipulada en el caso de uso. En los diagramas posteriores se identificarán con el nombre *E:Nombre*.
- **Clase de interfaz de usuario:** se utilizan para describir la interacción entre el sistema y sus actores. Suelen representar abstracciones de ventanas, interfaces de comunicación, formularios, etc. En los diagramas posteriores se identificarán con el nombre *I:Nombre*.
- **Clases de control:** son responsables de la coordinación, secuencia de transacciones y control de los objetos relacionados con un caso de uso. En los diagramas posteriores se identificarán con el nombre *C:Nombre*.

6.6.1.1. Descripción de la iteración de objetos

En esta tarea se expondrá los diagramas de iteración asociado a cada caso de uso definido con los objetos presentes y que han sido definidos en el apartado anterior.

El diagrama de interacción, representa la forma en como un cliente (Actor) u objetos (Clases) se comunican entre sí en petición a un evento.

A continuación se recogen al detalle los diagramas de interacción de cada caso de uso del sistema. En algunos casos se han realizado varios diagramas para un mismo caso para presentar la solución de una manera más clara y entendible.

- **CU-00: Acceso a la aplicación**
 - *Diagrama de secuencia CU-00_1:* Un usuario registrado se loguea.

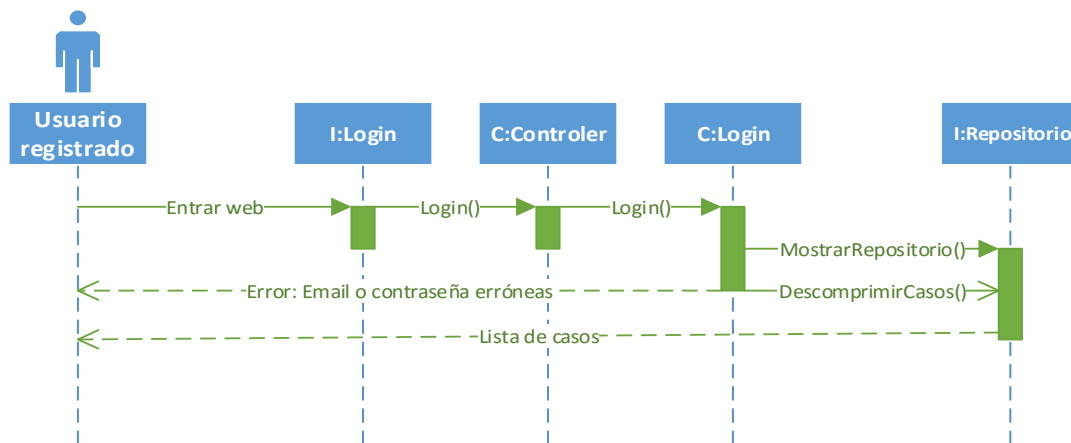


Ilustración 37 Diagrama de secuencia CU-00_1

- Diagrama de secuencia CU-00_2: Un usuario registrado recupera la contraseña.

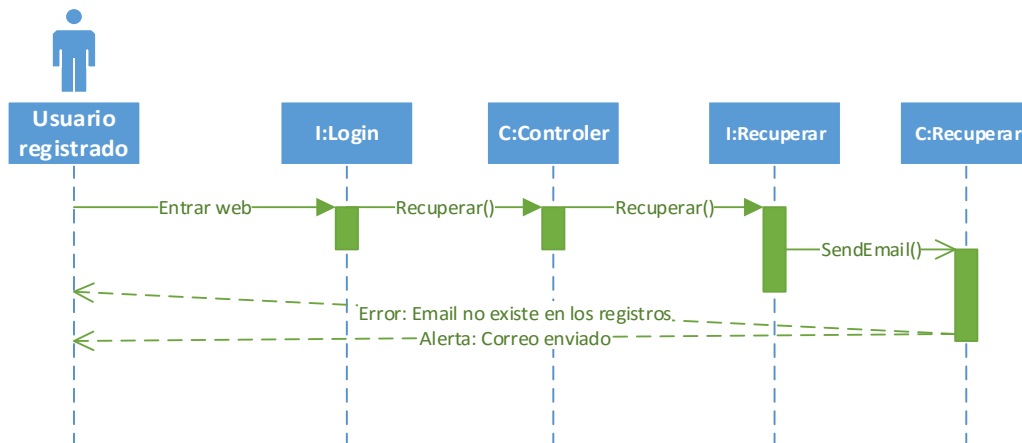


Ilustración 38 Diagrama de secuencia CU-00_2

- Diagrama de secuencia CU-00_3: Un usuario no registrado se registra.

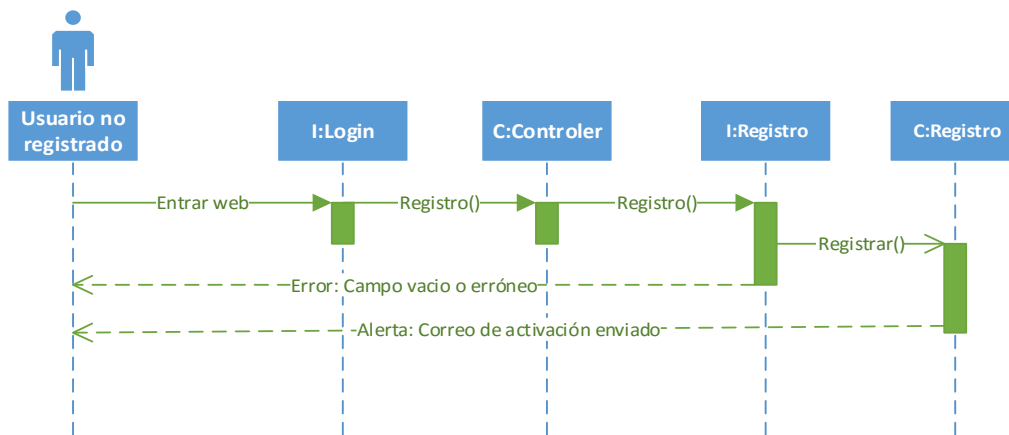


Ilustración 39 Diagrama de secuencia CU-00_3

- **CU-01: Cierre de sesión:** Usuario logueado cierra sesión.

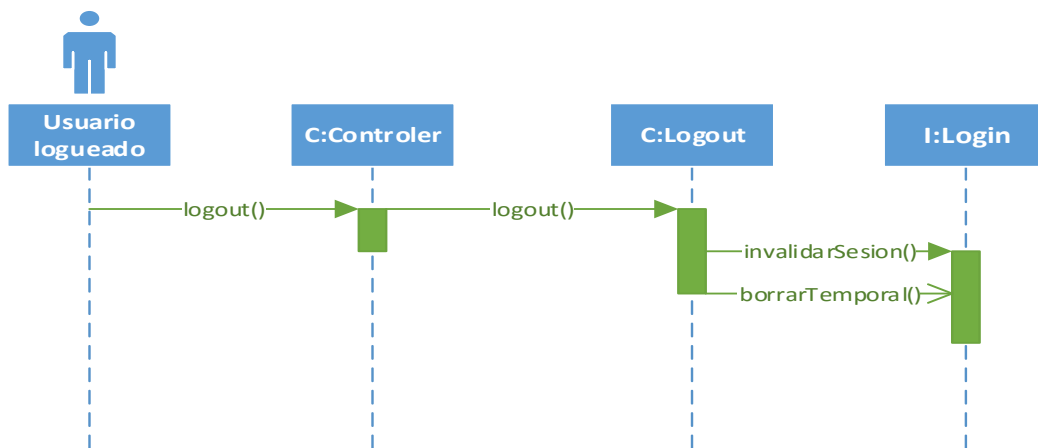


Ilustración 40 Diagrama de secuencia CU-01

- **CU-02: Gestión de cuenta:** Usuario logueado cambia datos de registro de su cuenta.

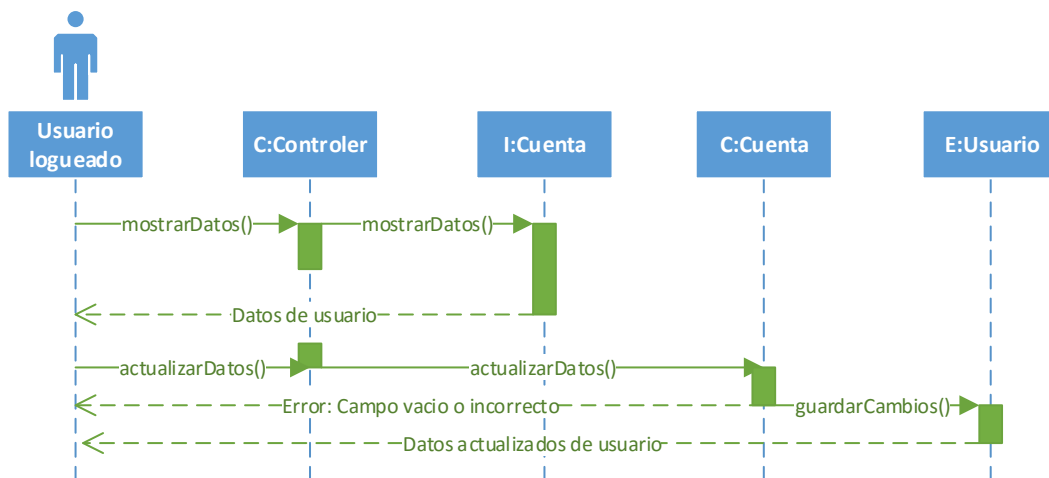


Ilustración 41 Diagrama de secuencia CU-02

- **CU-03: Subir caso de simulación:** Usuario logueado sube caso de simulación.

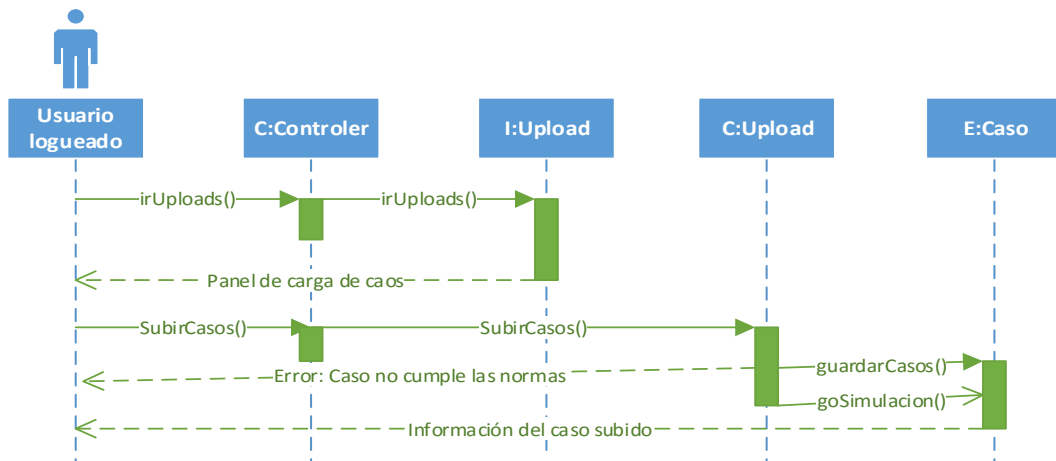


Ilustración 42 Diagrama de secuencia CU-03

- **CU-04: Gestión de casos**
 - *Diagrama de secuencia CU-04_1:* Usuario propietario o compartido visualiza la solución del caso.

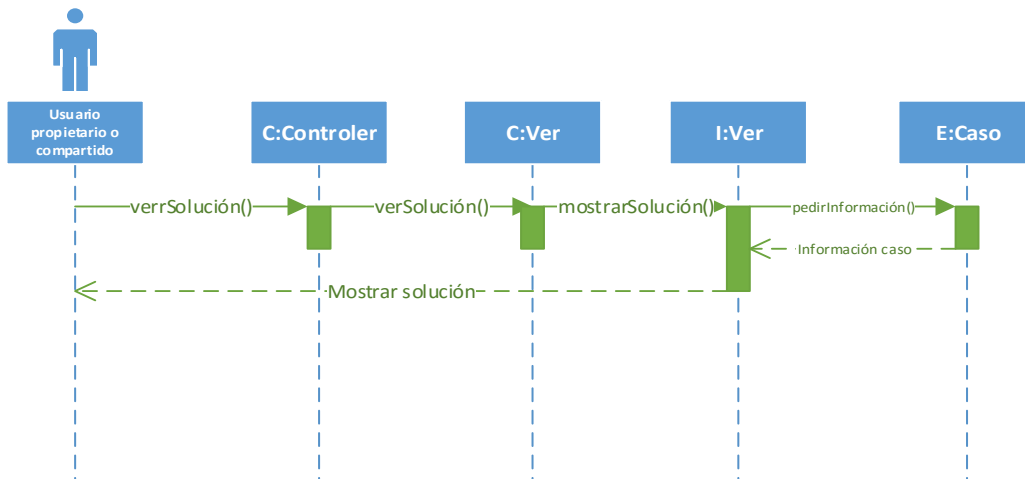


Ilustración 43 Diagrama de secuencia CU-04_1

- *Diagrama de secuencia CU-04_2:* Usuario propietario o compartido re-lanzan la simulación.

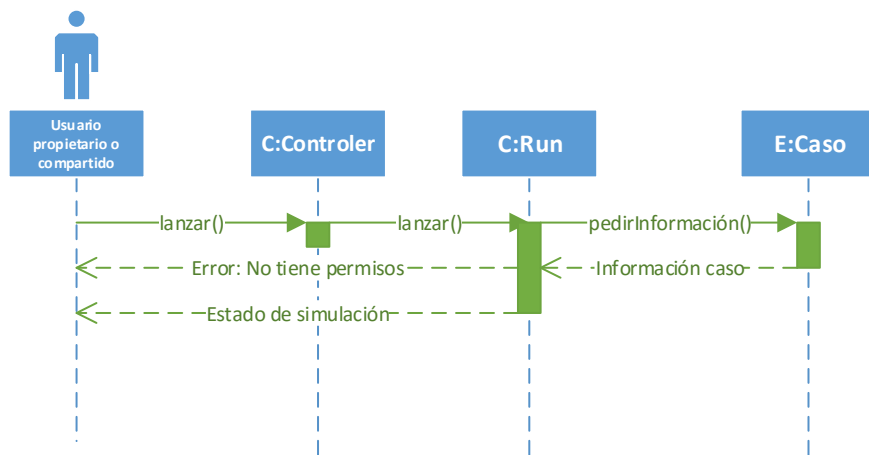


Ilustración 44 Diagrama de secuencia CU-04_2

- Diagrama de secuencia CU-04_3: Usuario propietario o compartido comparten un caso.

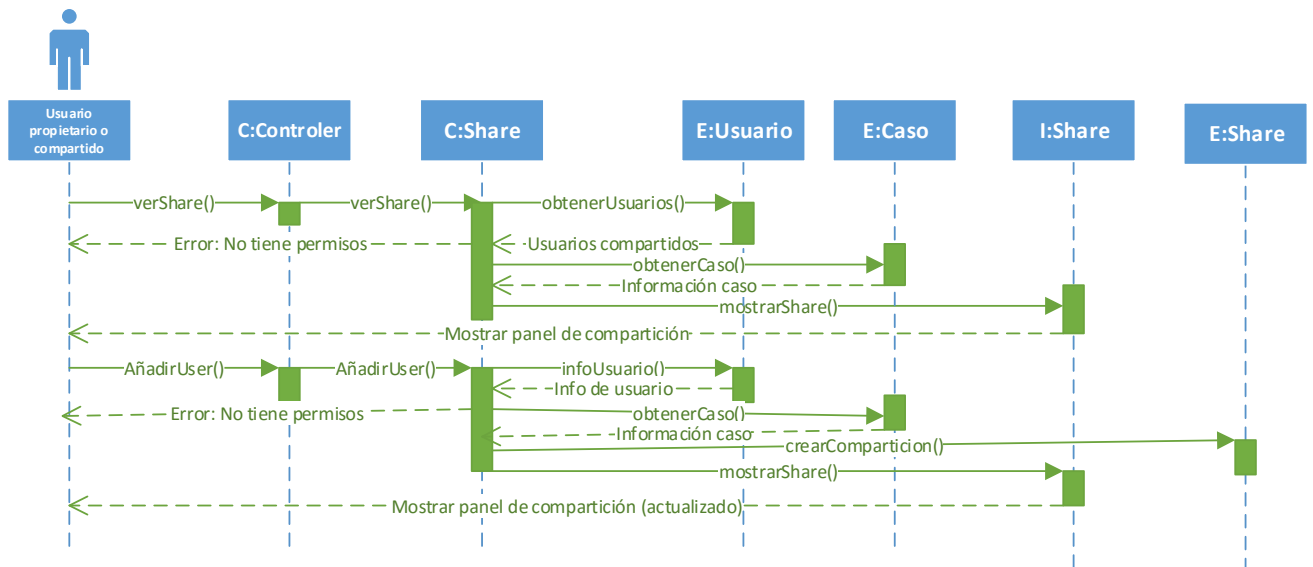


Ilustración 45 Diagrama de secuencia CU-04_3

- Diagrama de secuencia CU-04_4: Usuario propietario o compartido descargan un caso.

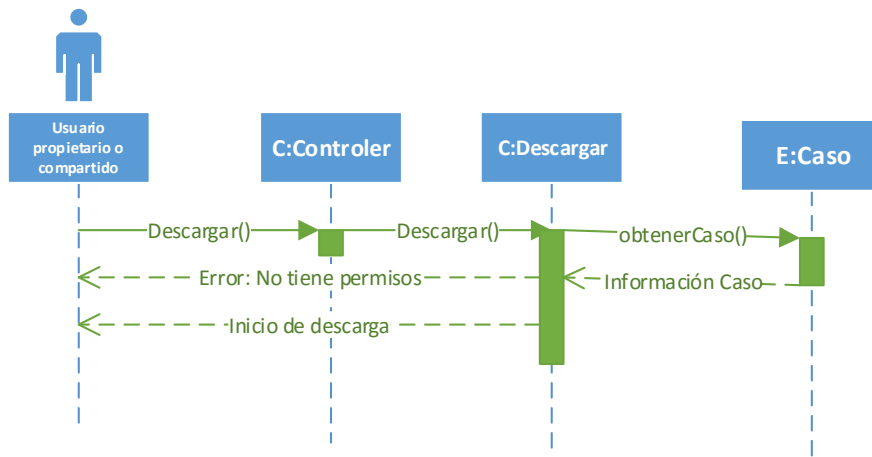


Ilustración 46 Diagrama de secuencia CU-04_4

- *Diagrama de secuencia CU-04_5: Usuario propietario o compartido borran un caso.*

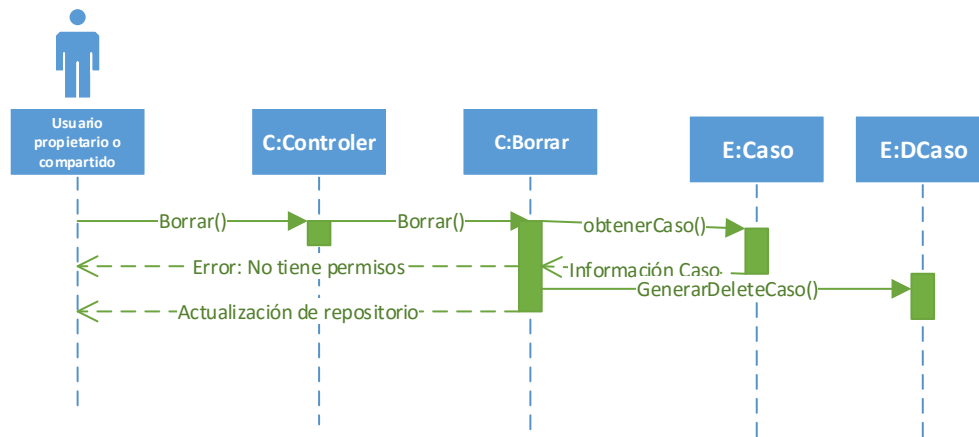


Ilustración 47 Diagrama de secuencia CU-04_5

- **CU-05: Gestión de permisos de caso:**
 - *Diagrama de secuencia CU-05_1: Usuario propietario o compartido añaden un usuario a la compartición.*

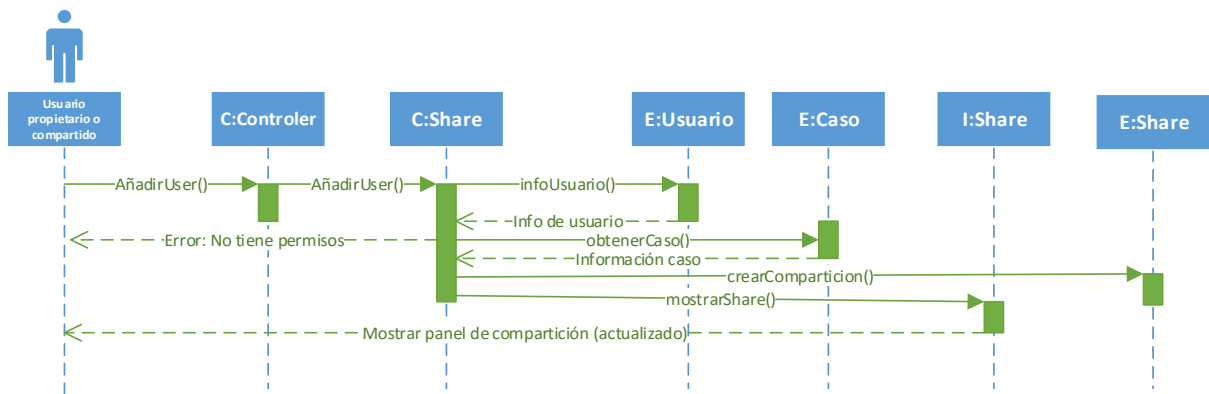


Ilustración 48 Diagrama de secuencia CU-5_1

- Diagrama de secuencia CU-5_2: Usuario propietario o compartido eliminan un usuario de la compartición.

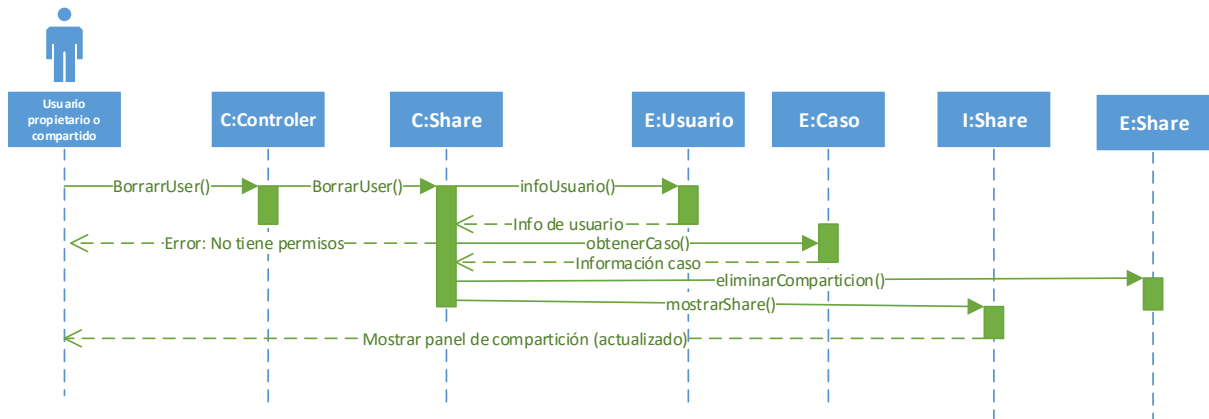


Ilustración 49 Diagrama de secuencia CU-5_2

- Diagrama de secuencia CU-5_3: Usuario propietario o compartido modifican permisos a un usuario.

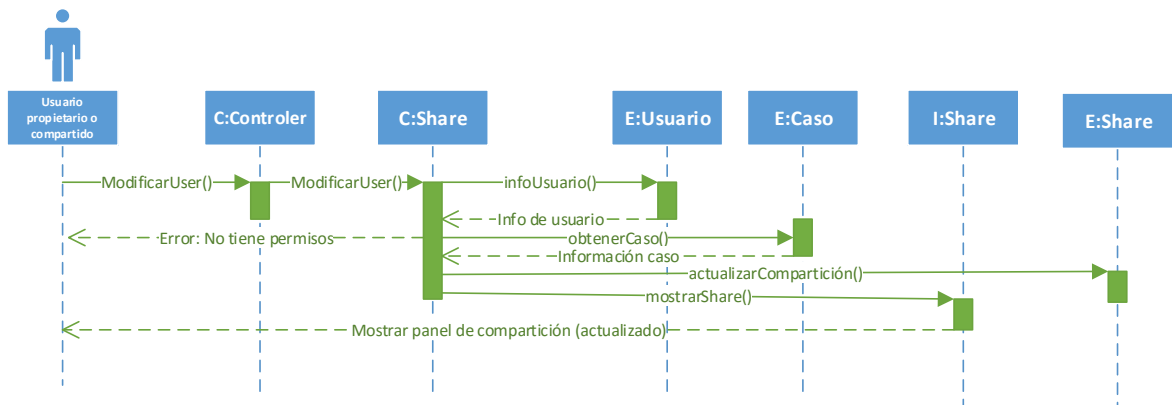


Ilustración 50 Diagrama de secuencia CU-05_3

- **CU-06: Elegir idioma:** Usuario elige idioma de la aplicación.

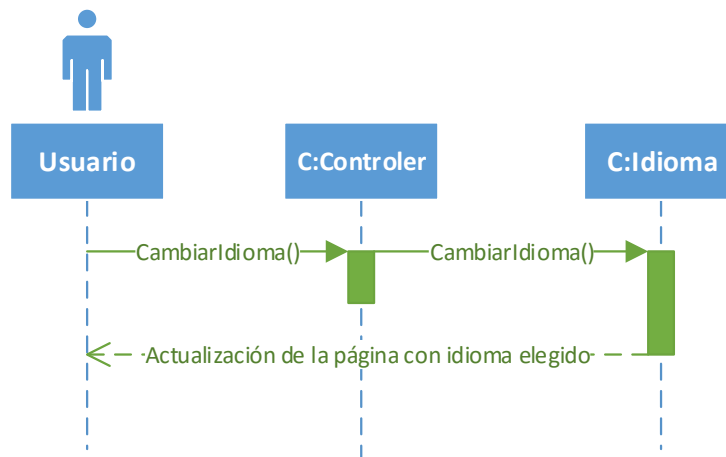


Ilustración 51 Diagrama de secuencia CU-06

- **CU-07: Navegación entre casos de simulación:** Usuario logueado navega entre las páginas de casos de su repositorio.

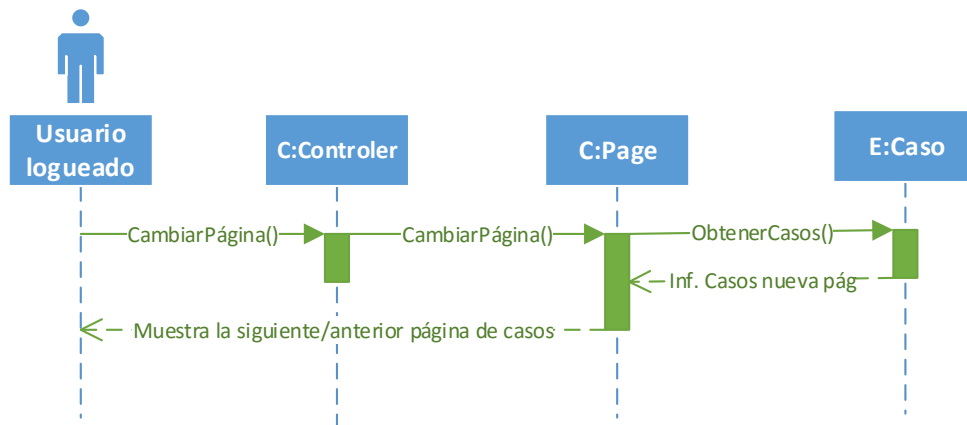


Ilustración 52 Diagrama de secuencia CU-07

6.7. Análisis de clases

En esta actividad se describen cada una de las clases que se han obtenido en la actividad anterior más las posibles que se puedan deducir de los requisitos. Se identifican las responsabilidades de cada clase, sus atributos y las relaciones entre ellas

En esta sección se identificarán todas las clases, se definirá su funcionalidad y atributos. Posteriormente se identificarán las asociaciones de estas con otras clases del sistema.

Antes de adentrarse en los detalles de cada una se proporciona un diagrama con las principales clases.

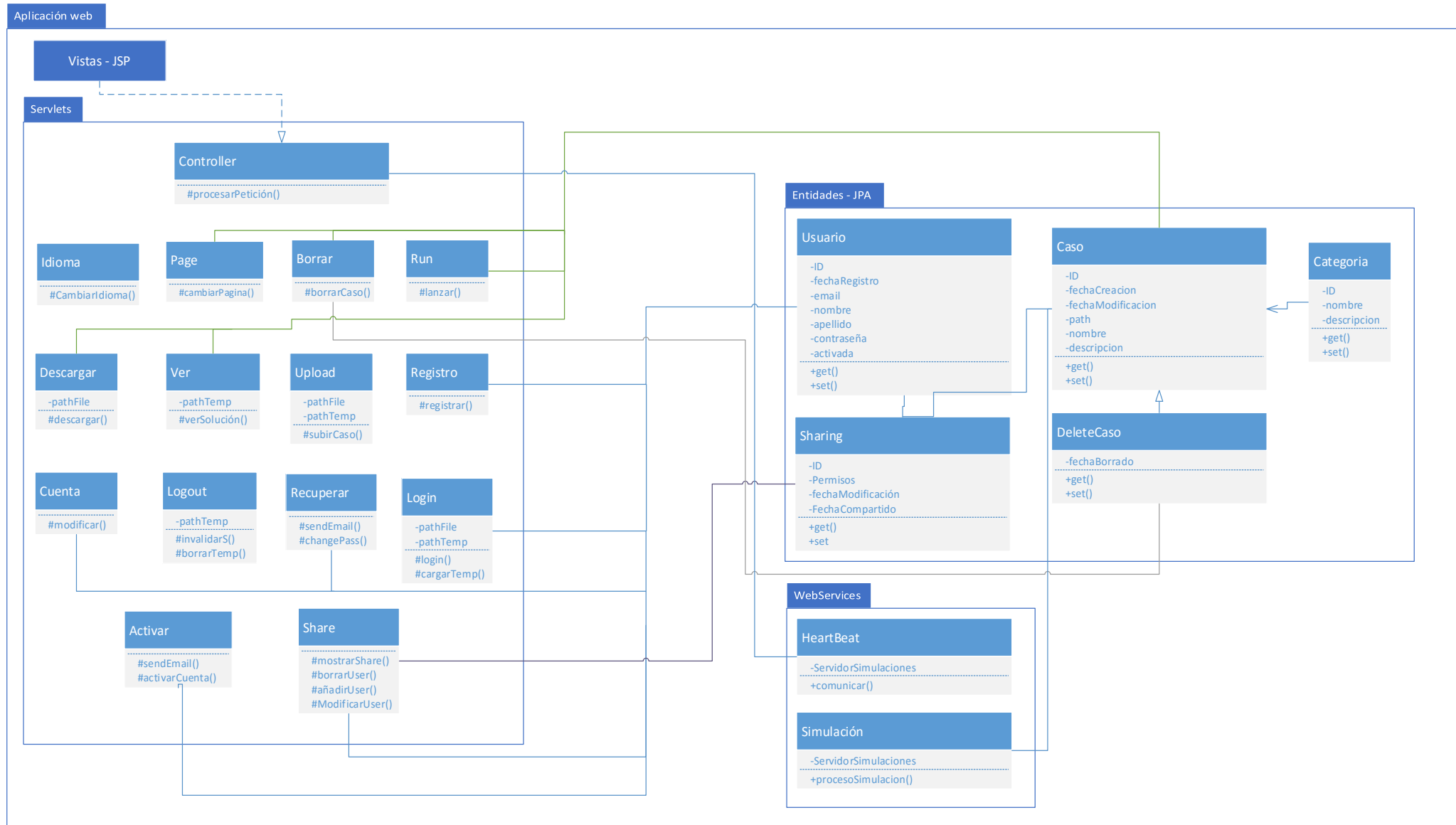


Ilustración 53 Análisis clases

Recordar que al ser esta la fase de análisis puede que no se contemplen en un primer momento todas las clases, será posteriormente en el diseño donde se dé una versión final. También aclarar, como ya se advirtió en la identificación de los subsistemas, que la lógica de negocio estará contenida en los propios servlets, por lo que para esta tarea se usará mayoritariamente un nombre identificativo de las clases y funciones (en vez de las funciones principales *doGet*, *doPost*,.. de estos tipos de clases), será en el diseño donde se entre más al detalle en las especificaciones.

6.7.1. Identificación de responsabilidades y atributos

Las responsabilidades definen la funcionalidad de una clase. En esta sección se describirán los atributos y funcionalidad de cada una de las clases recogidas en el diagrama de arriba. Toda esta información será recogida en una tabla por cada clase. La cual tendrá el siguiente aspecto:

Nombre de la clase	
Responsabilidades	Breve descripción de la finalidad de la clase
Atributos	<ul style="list-style-type: none"> • Atributo1: Descripción • Atributo2: Descripción ...
Operaciones	<ul style="list-style-type: none"> • Funcionalidad1: Descripción • Funcionalidad2: Descripción ...

Tabla 118 Responsabilidades y atributos tabla ejemplo

A continuación se muestra la tabla equivalente de responsabilidades, atributos y operaciones de cada una de las clases del sistema.

Controller	
Responsabilidades	Esta clase es la encargada de centralizar todas las peticiones que llegan de los usuarios y redirigirla al servlet encargado de su procesamiento.
Atributos	-
Operaciones	<ul style="list-style-type: none"> • procesarPetición(): Lee la petición del usuario y se encarga de redirigirla • Lanzar heartBeat: Cuando se despliega la aplicación esta función se lanza automáticamente para lanzar el proceso de comunicación de estado con el servidor de simulaciones.

Tabla 119 clase Controler

Idioma	
Responsabilidades	Esta clase es la encargada de cambiar el idioma actual de la aplicación por el seleccionado por el usuario
Atributos	-
Operaciones	<ul style="list-style-type: none"> • cambiarIdioma(): Lee la opción elegida por el usuario y cambia el idioma de la aplicación. Recarga la página en la que estaba el usuario en ese momento.

Tabla 120 clase Idioma

Page

Responsabilidades	Esta clase es la encargada de cambiar de página de casos dentro del repositorio de un usuario
Atributos	-
Operaciones	<ul style="list-style-type: none"> • cambiarPágina(): Comprueba si la solicitada es la página siguiente o anterior y carga el contenido de casos que debería tener.

Tabla 121 clase Page

Borrar	
Responsabilidades	Esta clase es la encargada de borrar un caso del repositorio de un usuario
Atributos	-
Operaciones	<ul style="list-style-type: none"> • borrarCaso(): Recibe la orden de borrar un caso específico y lo elimina de del servidor y la base de datos. El caso borrado y su información se almacenan en una carpeta y tabla específica para casos borrados.

Tabla 122 clase Borrar

Run	
Responsabilidades	Esta clase es la encargada de lanzar la simulación de un caso
Atributos	-
Operaciones	<ul style="list-style-type: none"> • lanzar(): Recibe un caso de simulación y manda al servidor de simulaciones la orden de lanzar la simulación

Tabla 123 clase Run

Descargar	
Responsabilidades	Esta clase es la encargada de realizar la descarga de un caso
Atributos	<ul style="list-style-type: none"> • pathFiles: Ruta donde están almacenados los casos en el servidor
Operaciones	<ul style="list-style-type: none"> • descargar(): Recibe la información del caso a descargar, lo busca en la carpeta y ejecuta el proceso de descarga en la máquina del cliente.

Tabla 124 clase Descargar

Ver	
Responsabilidades	Esta clase es la encargada de proporcionar al usuario el resultado de la simulación de un caso
Atributos	<ul style="list-style-type: none"> • pathTemp: Ruta donde están almacenados el HTML de los casos
Operaciones	<ul style="list-style-type: none"> • verSolución(): Devuelve al usuario la solución del caso solicitado por el usuario.

Tabla 125 clase Ver

Upload	
Responsabilidades	Esta clase es la encargada de gestionar la subida de los casos al servidor y lanzar su posterior simulación.
Atributos	<ul style="list-style-type: none"> • pathFiles: Ruta donde están almacenados los casos en el servidor • pathTemp: Ruta donde están almacenados el HTML de los casos
Operaciones	<ul style="list-style-type: none"> • subirCaso(): Recibe el archivo que quiere subir el usuario y lo almacena en el servidor junto a su información en la base

	<p>de datos, previa comprobación de cumplimiento de normas y estándares.</p> <ul style="list-style-type: none"> • Lanzar simulación: Tras la subida de un caso se solicita la simulación de este al servidor de simulaciones.
--	---

Tabla 126 clase Upload

Registro	
Responsabilidades	Esta clase es la encargada de gestionar el registro de un nuevo usuario en la aplicación.
Atributos	-
Operaciones	<ul style="list-style-type: none"> • registrar(): Recibe los datos del formulario rellenado por el usuario y crea y almacena un nuevo usuario en la base de datos.

Tabla 127 clase Registro

Cuenta	
Responsabilidades	Esta clase es la encargada de gestionar todo cambio en los datos de la cuenta de un usuario.
Atributos	-
Operaciones	<ul style="list-style-type: none"> • modificar(): Recibe unos datos a cambiar en la cuenta del usuario y procede a su actualización en la base de datos.

Tabla 128 clase Cuenta

Logout	
Responsabilidades	Esta clase es la encargada de gestionar todo cambio en los datos de la cuenta de un usuario.
Atributos	<ul style="list-style-type: none"> • pathTemp: Ruta donde están almacenados el HTML de los casos
Operaciones	<ul style="list-style-type: none"> • invalidar(): invalida la sesión del usuario desconectado. • borrarTemp(): borra todo el HTML extraído de los casos asociados a ese usuario.

Tabla 129 clase Logout

Recuperar	
Responsabilidades	Esta clase es la encargada de realizar la recuperación contraseña de la cuenta de un usuario.
Atributos	-
Operaciones	<ul style="list-style-type: none"> • sendEmail(): Envía un correo al email recibido con los pasos para resetear la contraseña. • changePass(): actualiza los datos del usuario en la base de datos con la nueva contraseña.

Tabla 130 clase Recuperar

Login	
Responsabilidades	Esta clase es la encargada de realizar el login en la aplicación de un usuario
Atributos	<ul style="list-style-type: none"> • pathFiles: Ruta donde están almacenados los casos en el servidor • pathTemp: Ruta donde están almacenados el HTML de los casos

Operaciones	<ul style="list-style-type: none"> • login(): Comprueba que los datos de acceso son correctos y loguea al usuario. • cargarTemp(): extrae el HTML de los casos asociados a ese usuario.
-------------	---

Tabla 131 clase Login

Activar	
Responsabilidades	Esta clase es la encargada de realizar la activación de cuenta de un usuario tras un registro.
Atributos	-
Operaciones	<ul style="list-style-type: none"> • sendEmail(): Envía un correo al email del usuario recién registrado con el link de activación. • activarCuenta(): activa la cuenta del usuario y autoriza su uso.

Tabla 132 clase Activar

Share	
Responsabilidades	Esta clase es la encargada toda la gestión de compartición y permisos de casos
Atributos	-
Operaciones	<ul style="list-style-type: none"> • mostrarShare(): Carga el panel de compartición con toda la información de las comparticiones de ese caso para el usuario que lo solicita. • borrarUser(): borra a un usuario de la compartición de ese caso. • añadirUser(): añade a la compartición del caso un nuevo usuario. • modificarUser(): modifica los permisos de un usuario sobre un caso.

Tabla 133 clase Share

Usuario	
Responsabilidades	Esta clase entidad es la encargada almacenar la información de un usuario
Atributos	<ul style="list-style-type: none"> • ID: identificador único del usuario • fechaRegistro • email • nombre • apellido • contraseña • activada: booleano que marca si la cuenta ha sido activada o no
Operaciones	<ul style="list-style-type: none"> • get(): método para obtener el valor de los atributos • set(): método para cambiar el valor de los atributos

Tabla 134 clase Usuario

Caso	
Responsabilidades	Esta clase entidad es la encargada almacenar la información de un caso
Atributos	<ul style="list-style-type: none"> • ID: identificador único del caso • fechaCreación

	<ul style="list-style-type: none"> • fechaModificación • path: localización en el servidor • nombre • descripción
Operaciones	<ul style="list-style-type: none"> • get(): método para obtener el valor de los atributos • set(): método para cambiar el valor de los atributos

Tabla 135 clase Caso

Categoría	
Responsabilidades	Esta clase entidad es la encargada almacenar la información de una categoría de estado
Atributos	<ul style="list-style-type: none"> • ID: identificador único del estado • nombre: nombre del estado {pendiente, calculando, simulado, errores} • descripción
Operaciones	<ul style="list-style-type: none"> • get(): método para obtener el valor de los atributos • set(): método para cambiar el valor de los atributos

Tabla 136 clase Categoría

Categoría	
Responsabilidades	Esta clase entidad es la encargada almacenar la información de una compartición
Atributos	<ul style="list-style-type: none"> • ID: identificador único de la compartición • permisos: permisos del compartido en el caso • fechaModificación: fecha del último cambio en los permisos • fechaCompartido
Operaciones	<ul style="list-style-type: none"> • get(): método para obtener el valor de los atributos • set(): método para cambiar el valor de los atributos

Tabla 137 clase Categoría

DeleteCaso	
Responsabilidades	Esta clase entidad es la encargada almacenar la información de un caso borrado
Atributos	<ul style="list-style-type: none"> • FechaBorrado: fecha en el que el caso fue eliminado
Operaciones	<ul style="list-style-type: none"> • get(): método para obtener el valor de los atributos • set(): método para cambiar el valor de los atributos

Tabla 138 clase DeleteCaso

HeartBeat	
Responsabilidades	Esta clase es la encargada del WebServices con el servidor de simulaciones para saber su estado
Atributos	<ul style="list-style-type: none"> • Servidor de simulaciones: URI del servidor de simulaciones
Operaciones	<ul style="list-style-type: none"> • comunicar(): método que mantendrá el contacto continuo con el servidor de simulaciones para saber su estado.

Tabla 139 clase HeartBeat

Simulación	
Responsabilidades	Esta clase es la encargada del WebServices con el servidor de simulaciones realizar la simulación de un caso

Atributos	<ul style="list-style-type: none"> • Servidor de simulaciones: URI del servidor de simulaciones
Operaciones	<ul style="list-style-type: none"> • procesoSimulación(): método que ejecuta toda las llamadas necesarias al servidor de simulaciones para realizar la simulación del caso correctamente.

Tabla 140 clase Simulación

6.7.2. Identificación de asociaciones y agregaciones

En esta tarea se estudian los mensajes que se han identificado entre los objetos de los diagramas de interacción, para determinar qué asociaciones existen entre las clases correspondientes. Toda esta información será recogida en una tabla por cada asociación. La cual tendrá el siguiente aspecto:

Clase_X $\leftarrow \rightarrow$ Clase_Y	
Tipo	Asociación o agregación
Clases (cardinalidad)	<ul style="list-style-type: none"> • Clase_X (1) • Clase_Y (0..*)
Descripción	Una Clase_Y pertenece a una única Clase_X, y una Clase_X puede tener varias Clase_Y.

Tabla 141 Asociaciones y agregaciones tabla ejemplo

A continuación se muestra la tabla de asociaciones y agregaciones de cada una de las relaciones de clases del sistema. Se presentan las asociaciones entre las clases entidades:

Usuario $\leftarrow \rightarrow$ Caso	
Tipo	Asociación
Clases (cardinalidad)	<ul style="list-style-type: none"> • Usuario (1) • Caso (0..*)
Descripción	Un caso pertenece a un único usuario, y un usuario puede tener varios o ningún caso.

Tabla 142 Asociación Usuario-Caso

Usuario $\leftarrow \rightarrow$ Sharing	
Tipo	Asociación
Clases (cardinalidad)	<ul style="list-style-type: none"> • Usuario (2) • Sharing (0..*)
Descripción	Una compartición (sharing) pertenece a sólo dos usuarios, y un usuario puede tener varias o ninguna compartición.

Tabla 143 Asociación Usuario-Share

Caso $\leftarrow \rightarrow$ Sharing	
Tipo	Asociación
Clases (cardinalidad)	<ul style="list-style-type: none"> • Caso (1) • Sharing (0..*)
Descripción	Una compartición (sharing) pertenece a sólo un caso, y un caso puede tener varias o ninguna compartición.

Tabla 144 Asociación Caso-Share

Usuario ← → DeleteCaso	
Tipo	Asociación
Clases (cardinalidad)	<ul style="list-style-type: none"> • Usuario (1) • DeleteCaso (0..*)
Descripción	Un caso borrado (DeleteCaso) pertenece a sólo un usuario, y un usuario puede tener varios o ningún caso borrado.

Tabla 145 Asociación Usuario-DeleteCaso

Caso ← → Categoría	
Tipo	Asociación
Clases (cardinalidad)	<ul style="list-style-type: none"> • Caso (0..*) • Categoría (1)
Descripción	Una categoría de estado (Categoría) pertenece a ningún o varios casos, y un caso puede tener un sólo estado.

Tabla 146 Caso-Categoría

6.7.3. Identificación de generalizaciones

El objetivo de esta tarea es representar una organización de las clases que permita una implementación sencilla de la herencia y una agrupación semántica de las diferentes clases.

Para este sistema las generalizaciones encontradas son las siguientes.

Caso	
Tipo	Generalización
Nivel	Superclase
Superclase/s	
Subclase/s	DeleteCaso
Descripción	Define el contenido y funcionalidades de un caso de simulación

Tabla 147 Generalización Caso

DeleteCaso	
Tipo	Generalización
Nivel	Subclase
Superclase/s	Caso
Subclase/s	
Descripción	Define la información de los casos de simulación borrados

Tabla 148 Generalización DeleteCaso

6.8. Elaboración del modelo de datos

En esta tarea se detalla el modelo de datos de la aplicación, es decir, toda la información que se almacenará en el sistema y que formará el punto de partida para la base de datos.

A continuación se representan los conceptos y las relaciones entre ellos. Este diagrama será el punto de partida para el diseño de la base de datos.

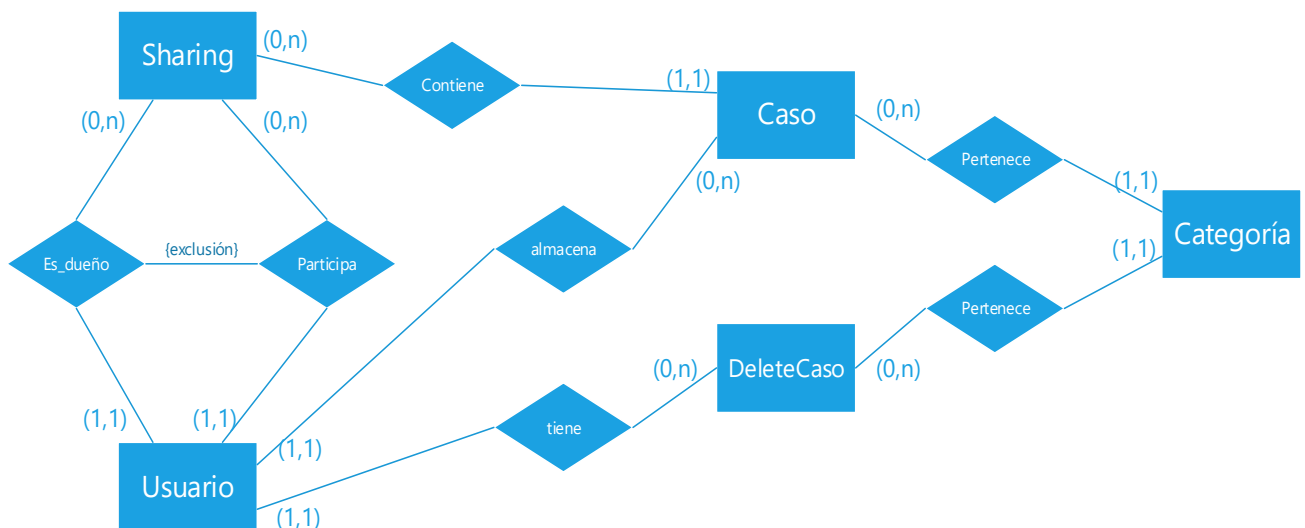


Ilustración 54 Modelo E/R de db

Los atributos de estas futuras tablas de la base de datos serán los mismos que los de las clases entidades equivalentes.

Los cuadrados son las entidades, los rombos las acciones y los números la cardinalidad de la relación.

6.8.1. Especificación de necesidades y de migración de datos y carga inicial

El sistema no necesitará una carga inicial de datos. Simplemente se configurará la base de datos inicialmente antes de las futuras inserciones con la creación de índices y disparadores para aumentar su rendimiento y consistencia.

6.9. Definición de interfaces de usuario

En este apartado se a exponer un primer boceto de las interfaces de usuario de la aplicación web. Se describirán sus principios generales, sus perfiles y diálogos, su comportamiento dinámico... Todas estas características buscan crear una interfaz que satisfagan los requisitos establecidos.

Se busca crear interfaces de fácil manejo, coherentes, flexibles y acorde a las necesidades del cliente y de los futuros usuarios de la aplicación.

En los próximos apartados también se identificarán los diferentes perfiles de usuario que interactuarán con el sistema y se determinará la naturaleza de los procesos que se llevan a cabo.

Finalmente se definirá el formato y contenido de las interfaces de pantalla especificando su comportamiento dinámico.

Para aclarar bien el contenido que tendrá este punto cabe destacar que en las tareas que componen este apartado se desarrollarán los siguientes pasos que posteriormente se completarán en proceso de diseño:

Paso 1. Conocer al usuario.

Paso 2. Entender la función de negocio en cuestión.

Paso 3. Aplicar principios de diseño de diálogos.

Paso 4. Seleccionar el tipo adecuado de ventanas.

Paso 5. Definir los menús del sistema.

Paso 6. Seleccionar los controles adecuados al tipo de interfaz.

Paso 7. Elegir los controles apropiados a la pantalla.

Paso 8. Organizar y presentar las ventanas.

Paso 9. Elegir los colores apropiados.

Paso 10. Crear iconos significativos.

Paso 11. Proporcionar mensajes y retroalimentación al resto de procesos.

Para finalizar resaltar que se está en fase de definición, en ningún caso se están añadiendo en análisis detalles de cómo se va a implementar la interfaz de usuario, salvo que supongan una restricción relevante, sino que se está describiendo cómo será dicha interfaz.

6.9.1. Especificación de principios generales de la interfaz

Para el diseño de la interfaz se tendrá en cuenta las necesidades, experiencia y capacidades de los futuros usuarios que interactuarán con la aplicación. Estos usuarios serán tenidos en cuenta en el diseño, ya que este no puede ser realizado exclusivamente a partir de las abstracciones realizadas en pasos anteriores del ciclo de vida.

Se tendrán en cuenta las limitaciones que presentan los usuarios a la hora de interactuar con una aplicación, una de las más importantes es el rechazo a la hora de reconocer conceptos nuevos. También se tendrán en cuenta que existen conceptos comunes y convencionales que son aceptados inconscientemente por el usuario (iconos, colocación de menús...), la utilización de estos conceptos en la interfaz facilitará su manejo por los usuarios.

El diseño de la interfaz se centrará en el conocimiento del usuario final y de sus capacidades, ya que el sistema se centrará en solucionar problemas al usuario, implicándole para ello en la tarea de diseño de la interfaz.

Las principales características que deben tener las interfaces de la aplicación web son las siguientes:

- **Claridad:** Se debe crear una interfaz entendible para el usuario y evitar en todo lo posible las ambigüedades.

- **Compleitud:** La interfaz debe ofrecer todas las funcionalidades establecidas en los requisitos.
- **Sencillez:** El diseño de la interfaz debe ser lo más minimalista posible. Cuyo principal objetivo es hacer la interfaz lo más intuitiva posible, de esta forma se conseguirá evitar que el usuario se sienta abrumado por el exceso de contenido y este pueda manejarse tranquilamente y sabiendo lo que está haciendo en cada paso.
- **Idioma:** Los textos de las interfaces se mostrarán por defecto en castellano. Pudiendo cambiarlo al inglés, tal y como se estableció en los requisitos.
- **Tecnología de la interfaz:** Como se explicó y estudió en 4.4.1. *Alternativa 1: Java Servlets + MySQL*, la tecnología usada será Java Servlets combinada con el uso de MySQL.
- **Restablecimiento:** Se incluirán mecanismos que permiten al usuario retornar a estados anteriores para corregir posibles errores propios. Para minimizar los errores cometidos por los usuarios finales cuando operan con el sistema, se incluirán medios para que el usuario pueda rehacerse de dichos errores (confirmación de acciones destructivas, implementar facilidades de deshacer,...).
- **Ayuda:** El usuario dispondrá de sistemas de guía y ayuda sensible al contexto. Para ello se integrarán en la interfaz las ayudas y guías necesarias a diferentes niveles, desde avisos y comunicación de las acciones hasta textos emergentes con la funcionalidad de los iconos.
- **Poca profundidad:** las diferentes páginas de la aplicación deben ser accesibles sin tener que recorrer enrevesados caminos.

6.9.2. Identificación de perfiles y diálogos

En base al apartado 6.4.1. *Especificación de casos de uso* se resumen los perfiles en la siguiente tabla:

Perfil	Descripción
Usuario logueado	Cualquier usuario que interactúe con la aplicación estando logueado
Usuario no logueado	Cualquier usuario que interactúe con la aplicación sin estar logueado
Usuario propietario	Cualquier usuario que interactúe con un caso de la aplicación siendo él su propietario
Usuario compartido	Cualquier usuario que interactúe con un caso de la aplicación sin ser él el propietario

Tabla 149 Perfiles y diálogos

6.9.3. Especificación de formatos individuales y de la interfaz de pantalla

En este punto se especifican el formato individual de cada pantalla, desde el punto de vista estático, es decir, elementos que van a aparecer siempre en pantalla de la misma forma.

Las pantallas consideradas para la aplicación web a desarrollar son las siguientes:

- Página de inicio.
- Menú
- Repositorio
- Gestión de cuenta
- Subida de casos
- Panel de compartición
- Vista de solución
- Error/Información

A continuación se pasa a explicar en detalle cada una de las pantallas anteriores, así como la representación del diseño de estas con un sencillo boceto.

6.9.3.1. *UI-00: Página de inicio*

Compuesta por una sencilla ventana de login colocada centralmente (como lo estarán todas de la aplicación). Esta ventana de login estará compuesta por un formulario con dos campos que será de *email*(1) y *contraseña*(2), los cuales deberán rellenarse para acceder a la aplicación a través del botón de *login*(3).

Esta ventana también proporcionará dos links que redirigirán a la pantalla de *registro*(4) y *recuperación de contraseña*(5) respectivamente. La primera estará compuesta por un formulario que pedirá al usuario *email*, *contraseña*, *nombre* y *apellidos*. Por otro lado la segunda estará compuesta por un formulario con un campo que pedirá el *email* de la cuenta de la que se quiere recuperar la contraseña. En ambas pantallas se podrá volver al login mediante el link de *volver*(6).

E-mail (1)

Contaseña (2)

[¿No recuerdas tu contraseña?](#) (5)

[Registrarse](#) (4) (3)

Ilustración 55 IU-00(1)

(4) **Formulario de registro**

E-mail

Nombre

Apellidos

Contraseña

[Volver](#) (6)

Ilustración 56 IU-00(2)

(5)

Recuperación de contraseña

Texto informativo para el usuario sobre los pasos a seguir para recuperar la contraseña

E-mail

[Volver \(6\)](#)

Ilustración 57 IU-00(3)

6.9.3.2. UI-01: Menú

El menú estará compuesto por el *selector de idioma*(1), el *estado del servidor de simulaciones*(2), *nombre y apellido del usuario* (3), *link a gestión de cuenta*(4) y *link a logout* (5). Este menú estará colocado en la parte superior de la pantalla. La opción (1) será visible en todas las pantallas de la aplicación, mientras que el resto únicamente para pantallas en las que para acceder sea necesario un login previo.

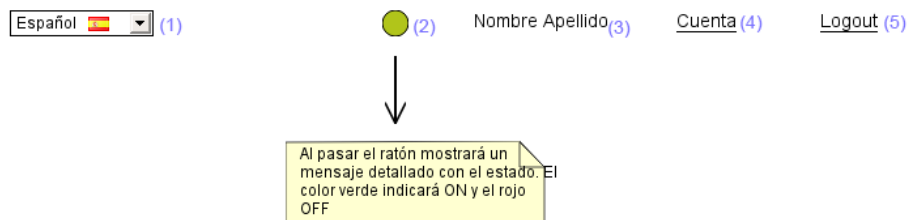


Ilustración 58 IU-01

6.9.3.3. UI-02: Repositorio

Esta pantalla mostrará los *casos*(1) a los que tiene acceso un usuario. Estará compuesta por una ventana central en la que cada caso y todos sus *iconos de funcionalidades*(2) ocuparán una línea. Se mostrarán a lo sumo cuatro casos propios y cuatro compartidos por cada ventana. En caso de que un usuario supere esta cifra se mostrará un link inferior para ir a la *página siguiente/anterior*(3).

Esta pantalla también mostrará un icono para *subir nuevos casos*(4), que estará en la parte superior junto al título de la ventana. Todos los iconos del repositorio al pasarles el ratón por encima mostrarán un pequeño título identificativo con su funcionalidad.



Ilustración 59 UI-02

6.9.3.4. UI-03: Gestión de cuenta

Esta pantalla será con la que interactuará el usuario para modificar sus datos de registro. En ella podrá modificar mediante un formulario *nombre*(1), *apellidos*(2) y *contraseña*(3). El *email*(4) no podrá ser modificado por el usuario.

Para actualizar los cambios habrá que pulsar en el botón *guardar*(5) y para volver al repositorio en el icono de *atrás*(6) colocado a la derecha del título.

Ilustración 60 UI-03

6.9.3.5. UI-04: Subida de casos

Se mostrará al usuario una ventana con una *zona(1)* en la que podrá arrastrar todos los casos que quiera subir o simplemente al hacer click en ella poder subirlos de forma tradicional. En el momento en el que un archivo sea puesto en la zona comenzará automáticamente su subida.

Esta pantalla también contará con un icono a la derecha del título para volver *atrás(2)* al repositorio.

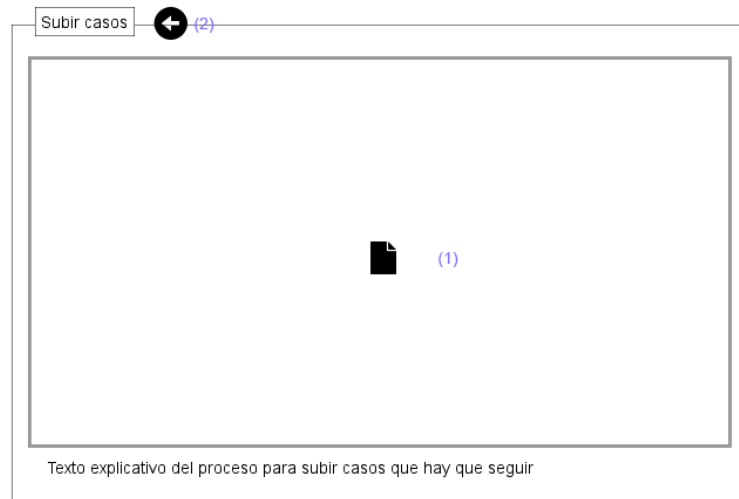


Ilustración 61 UI-04

6.9.3.6. UI-05: Panel de compartición

Será la pantalla que se muestre tras elegir la opción compartir de un proyecto. Esta pantalla cuenta con un *formulario(1)* para introducir el email del usuario con el que se quiere crear una compartición del caso.

En su parte inferior muestra una *tabla(2)* en la que el usuario podrá gestionar todos los permisos de los usuarios a los que él dio acceso al caso. Finalmente contará con un botón para *guardar(3)* los cambios en los permisos y, como en las pantallas anteriores, habrá un icono para *volver(4)* al repositorio.

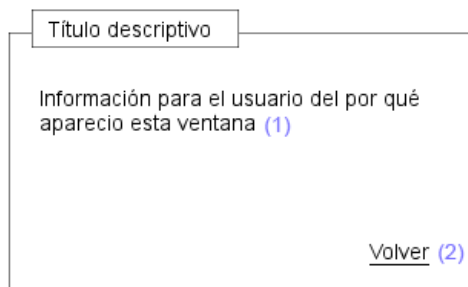


Ilustración 64 UI-07

6.9.4. Especificación del comportamiento dinámico de la interfaz

El objetivo de esta tarea es definir los flujos entre los distintos formatos de interfaz de pantalla descritos anteriormente, y también dentro del propio formato.

Para ello se usará un mapa de navegación en el que el significado de los símbolos usados es el siguiente.

Símbolo	Significado
	Página de inicio o página de entrada a la aplicación
<div>Página</div>	Página independiente de la aplicación web.
<div>Pop-up</div>	Página pop-up, ventana de dialogo o ventana emergente
	Navegación bidireccional
	Navegación unidireccional

Tabla 150 Símbolos mapa de navegación

A continuación se adjunta el mapa de navegación considerado para esta aplicación web a desarrollar. El formato de nombre de las páginas del esquema es la identificación de la interfaz de usuario.

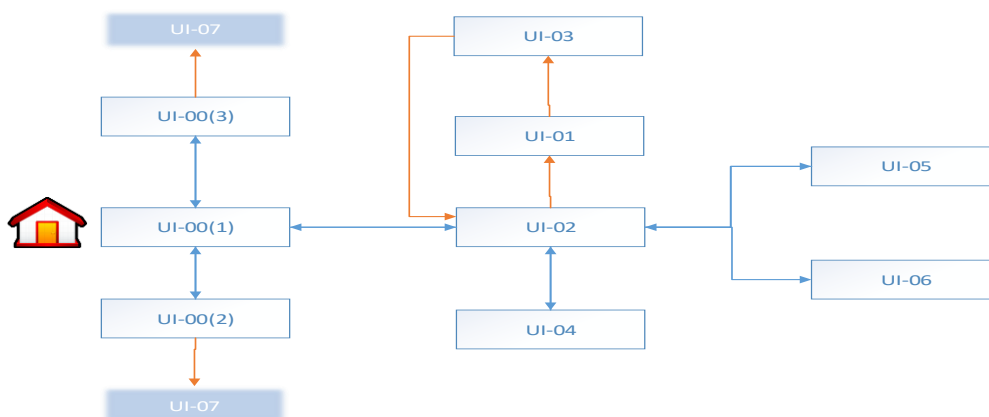


Ilustración 65 Mapa de navegación

6.9.5. Especificación de formatos de impresión

La aplicación ni genera salidas impresas, ni acepta entradas impresas. Únicamente aceptará como entrada archivos de subida y generará como salida la descarga de casos de simulación.

La aplicación no se responsabiliza del uso de la impresión de las interfaces de la aplicación por parte de un usuario con herramientas ajenas a esta.

6.10. Análisis de consistencia y especificación de requisitos

El objetivo de esta tarea es asegurar la calidad de los distintos modelos que se han generado a lo largo del proceso de análisis, y verificar que las partes implicadas en el proyecto tienen el mismo concepto del sistema.

6.10.1. Verificación y análisis de consistencia entre modelos

El objetivo de esta tarea es asegurar que los distintos modelos son correctos formalmente, es decir, que se han realizado de acuerdo con las normas que se han establecido en 6.3.3. *Especificación de estándares y normas.*

Se comprobará que los modelos generados son coherentes entre sí, comprobando que no existen ambigüedades ni duplicación de información. Por lo que las diferentes condiciones que deben cumplir son las siguientes:

- Consistencia entre clases y casos de uso.
 - Toda clase tendrá como mínimo un caso de uso asignado. Pero no habrá máximo en casos de uso asignados a una clase.
 - Todo caso de uso debe tener una clase asignada, sin máximo en la cantidad de clases asignadas a un caso de uso.
 - Los subsistemas definidos satisfacen la realización de todos los casos de uso.
 - Cada mensaje entre objetos se corresponde con una operación de una clase, y que todos los mensajes se envían a la clase correcta.
- Consistencia entre requisitos de usuario y requisitos software.
 - Todo requisito software debe tener un requisito de usuario asignado. Sin máximo de requisitos de usuario asignados.
 - Todo requisito de usuario debe tener un requisito software asignado. Sin máximo de requisitos software asignados.
- Consistencia entre casos de uso e interfaces de usuario.
 - Todo caso de uso debe tener una interfaz de usuario asignada. Sin máximo de casos de uso asignados a una interfaz.

Para comprobar que todas estas pautas se cumplen se han introducido diagramas de secuencia para la comprobación de la consistencia entre clases y casos de uso, los cuales pueden ser visualizados en el punto 6.6.1. *Identificación de clases asociadas a un caso de uso.*

Para el segundo punto, consistencia entre requisitos de usuario y software, se ha realizado una matriz de trazabilidad que puede ser consultada en el punto 6.4.2.9. *Matriz de trazabilidad: UR-SR*.

Por último, para comprobar la consistencia entre casos de uso e interfaces de usuario se adjunta una matriz de trazabilidad a continuación.

	UI-00	UI-01	UI-02	UI-03	UI-04	UI-05	UI-06	UI-07
CU-00	X							X
CU-01		X						
CU-02		X		X				
CU-03			X		X			
CU-04			X				X	
CU-05						X		
CU-06		X						
CU-07			X					X

Tabla 151 Matriz de trazabilidad CU-UI

6.10.2. Validación de los modelos

Mediante las matrices de trazabilidad y diagramas de secuencia se puede comprobar que todos y cada uno de los requisitos, casos de uso, clases e interfaces están relacionadas. La falta de relación en alguno implicaría que no se ha usado o tenido en cuenta algún requisito de usuario en el análisis o que hay necesidad de crear un nuevo requisito de usuario para subsanar ese fallo.

El coste de un error u omisión en el análisis, crece exponencialmente conforme se van realizando todas las fases del desarrollo, pudiendo llegar a tener consecuencias graves si el fallo se detecta cuando el sistema ha entrado en la fase explotación.

7. Diseño del sistema

Este apartado de la memoria se centra en el diseño del sistema de información a desarrollar en este TFG.

7.1. Objetivo

El principal objetivo de esta fase es resolver el problema que se ha descrito y modelado en la fase del análisis del sistema. Con esta solución se busca definir completamente la arquitectura del sistema y su entorno tecnológico que le va a proporcionar soporte.

En base a la información anterior se generará todas las especificaciones necesarias para la construcción del sistema, la descripción técnica del plan de pruebas y la definición de requisitos de implantación del sistema.

La importancia de esta tarea es clave para facilitar enormemente el desarrollo y posterior implantación del sistema, ya que el diseño proporciona el esqueleto del sistema y la forma en que los diferentes componentes de él interactúan.

7.2. Alcance

Una de las primeras tareas del diseño es determinar la arquitectura que alcance los objetivos del sistema. Seguidamente se determinarán los mecanismos genéricos que servirán de guía para el diseño detallado del sistema de información.

Por otro lado también se especificarán casos de uso reales con el propósito de determinar el comportamiento del sistema para cada uno mediante objetos o subsistemas de diseño que interactúan.

En el diseño también se transformará el modelo de clases previo especificado en el análisis en el punto 6.7. *Análisis de clases* en un modelo de clases de diseño. Este modelo recogerá una especificación detallada de cada una de las clases, sus atributos, métodos y relaciones establecidas entre ellas.

Por ultimo pueden englobarse las actividades encaminadas a completar el diseño del sistema de información. Para ello, se generaran las especificaciones de construcción, se establecerá la especificación técnica del plan de pruebas y se elaboraran los requisitos de implantación, actividad encaminada a completar el catálogo de requisitos con aquellos relacionados con la documentación que el usuario requiere para operar con el nuevo sistema, y los relativos a la propia implantación del sistema en el entorno de operación.

7.3. Definición de la arquitectura del sistema

En esta tarea se especifica el particionamiento físico del sistema, es decir, los equipos (nodos) en los que va a funcionar el sistema y las comunicaciones entre los mismos.

7.3.1. Definición de niveles de arquitectura

En esta tarea se definen los nodos y comunicaciones entre nodos que definen las particiones físicas del sistema.

La arquitectura a seguir es la que se ha establecido en el análisis, concretamente en el punto 6.5. *Identificación de subsistema de análisis*. La información de ese apartado del análisis será la base para este punto.

La arquitectura del sistema que se seguirá es la proporcionada del metapatrón que surge de la unión del patrón arquitectónico MVC (Model View Controller) y el patrón Front-Controller.

A modo resumen de lo explicado en el análisis, se ha optado por la arquitectura MVC por su sencillez, su reducción de dependencias, su reutilización y por la flexibilidad que proporciona ante futuros cambios.

Por último se adjuntan las dos imágenes que se usaron en análisis para describir los niveles de la arquitectura elegida.

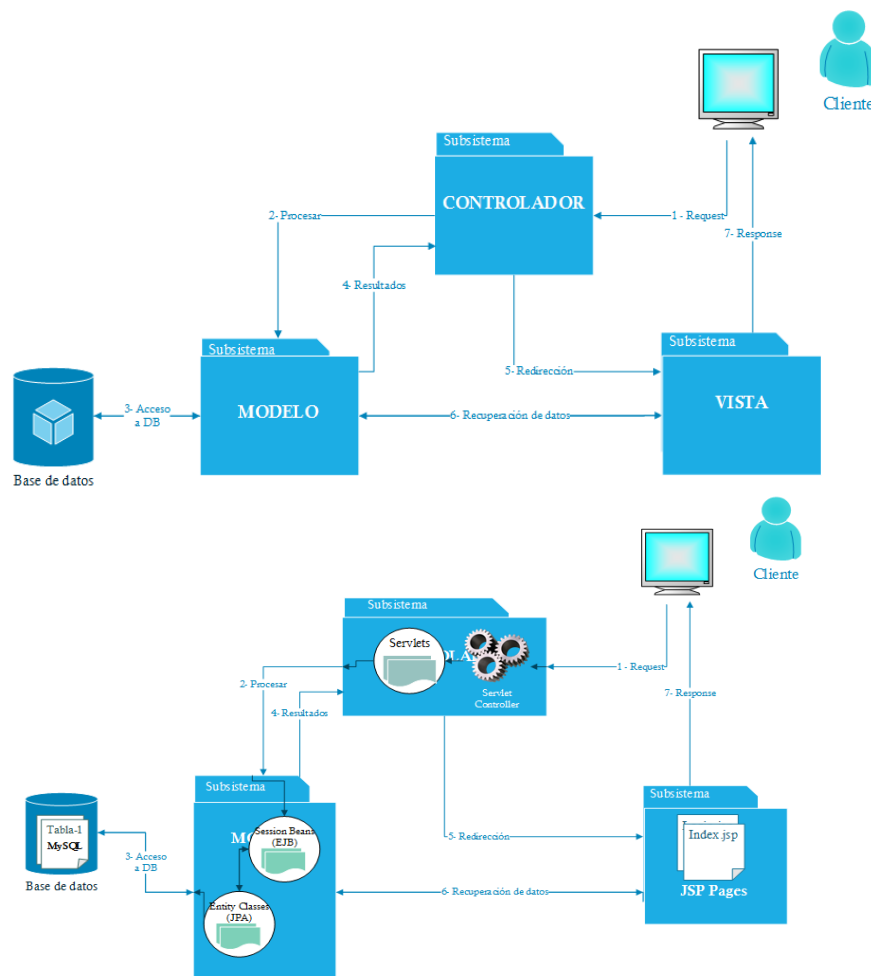


Ilustración 66 Arquitectura del sistema

7.3.2. Especificación de excepciones

En esta tarea se definen situaciones anómalas que se puedan producir en el funcionamiento y ejecución del sistema de información, como por ejemplo intentar acceder a un elemento de datos que no existe, o la caída del sistema de comunicaciones durante el funcionamiento del sistema.

Para enumerar las excepciones posibles se usarán tablas con los siguientes atributos cuyo objetivo será describir con detalle la excepción.

- Identificador y nombre.
- Tipo y descripción de la excepción.
- Estado en el que se encuentra el sistema antes de producirse la excepción.
- Elemento afectado.
- Respuesta del sistema de información.
- Elemento asociado a la respuesta esperada del sistema.

EX-XX Nombre	
Tipo y descripción	
Estado anterior	
Elemento afectado	
Respuesta del sistema	
Elemento asociado a la respuesta	

Tabla 152 Excepción plantilla

A continuación se presentan las posibles excepciones que pueden darse en el funcionamiento y ejecución del sistema.

EX-00 Formulario incorrecto	
Tipo y descripción	E/S. El usuario deja campos vacíos o incorrectos en un formulario de la aplicación
Estado anterior	Usuario acaba de terminar de rellenar el formulario y lo ha enviado para que se procese
Elemento afectado	Página JSP - Vista
Respuesta del sistema	El script de comprobación de campos del JSP muestra un mensaje al usuario del motivo del error
Elemento asociado a la respuesta	Base de datos Controlador

Tabla 153 EX-00

EX-01 Servidor de aplicación caído o inaccesible	
Tipo y descripción	Físico. El servidor donde está alojada la aplicación se ha caído o está inaccesible
Estado anterior	El servidor estaba operativo
Elemento afectado	Todo el sistema

Respuesta del sistema	El sistema intenta reiniciarse. Se envía un email al administrador web en caso de que el problema persista
Elemento asociado a la respuesta	Ninguno

Tabla 154 EX-01

EX-02 Servidor de simulaciones caído o inaccesible	
Tipo y descripción	Físico. El servidor de simulaciones se ha caído o está inaccesible
Estado anterior	El servidor de simulaciones estaba operativo
Elemento afectado	Partes del sistema que se comunican con él
Respuesta del sistema	La aplicación web intenta establecer comunicación con el servidor de simulaciones. Se informa a los usuarios activos en la aplicación de que el servidor de simulaciones está inoperativo. Si el problema persiste se manda un email al administrador web
Elemento asociado a la respuesta	Ninguno

Tabla 155 EX-02

EX-03 Base de datos caída o inaccesible	
Tipo y descripción	Físico. La base de datos con la que se comunica la aplicación no está operativa
Estado anterior	La base de datos estaba operativa
Elemento afectado	Todo el sistema
Respuesta del sistema	La aplicación trata de establecer conexión y genera mensajes de errores internos por cada comunicación que se intente hacer con la base de datos
Elemento asociado a la respuesta	Base de datos

Tabla 156 EX-03

EX-04 Base de datos llena	
Tipo y descripción	Memoria. La memoria física de la base de datos está llena imposibilitando las inserciones en ella
Estado anterior	Inserción en la base de datos
Elemento afectado	Partes del sistema que requieran de inserciones en la base de datos para funcionar
Respuesta del sistema	La base de datos impide hacer inserciones en ella y devuelve mensajes de error
Elemento asociado a la respuesta	Base de datos

Tabla 157 EX-04

EX-05 Servidor lleno	
Tipo y descripción	Memoria. La memoria física del servidor donde está alojada la aplicación está llena

Estado anterior	Usuario o sistema intentan almacenar algo en el servidor
Elemento afectado	Gestión de casos
Respuesta del sistema	El sistema devuelve mensajes de error en acciones que impliquen alojar algo en el servidor
Elemento asociado a la respuesta	Vista

Tabla 158 EX-05

7.3.3. Especificación de estándares y normas de diseño y construcción

En esta tarea se definen los estándares técnicos y de nomenclatura, normas y recomendaciones, que se deben emplear en el diseño y la construcción del sistema. A parte de las recogidas en el apartado del análisis 6.3.3. *Especificación de estándares y normas* se seguirán las de los siguientes párrafos.

Dentro de las normas de codificación se seguirá el convenio de Java para el nombrado de clases, variables, métodos,... Dentro de los servlets el nombre de estos seguirá el siguiente formato *NombreServlet.java*, donde el *Nombre* deberá ser una palabra simple identificativa.

En cuanto a los ficheros que se subirán al servidor, los casos de uso, aparte de las normas recogidas en el análisis se asignará como nombre del fichero la ID que se le asigne en su inserción en la base de datos.

Como ya se ha mencionado en varias ocasiones en esta memoria el idioma por defecto será el castellano. Dentro de la aplicación se podrá elegir entre castellano e inglés.

7.3.4. Identificación de subsistemas de diseño

En esta tarea se va a dividir de una forma lógica el sistema de información en subsistemas de diseño. Al subdividir el sistema se pretende reducir la complejidad y facilitar el mantenimiento. Para ello se parte de la división en subsistemas realizada en el punto 6.5. *Determinación de subsistemas de análisis* del análisis.

En esta tarea se realizará una descripción más detallada en términos de diseño de los diferentes subsistemas de la aplicación que han derivado de elegir el patrón de arquitectura MVC.

El subsistema vista (view) incluirá todos los ficheros JSP con los que el usuario interactuará para comunicarse con el sistema. La funcionalidad de este sistema es recoger la información introducida o solicitada por un usuario para transferírsela al servlet controlador que será el encargado de gestionar esta información y enviar posteriormente el resultado para ser mostrado en la vista.

El subsistema controlador (controller) será el contenedor de los servlets encargados de recoger las peticiones que realice el usuario a través de las páginas de la vista. Estas peticiones

serán atendidas primero por el servlet controller que será el que la redirija al servler encargado que contenga la lógica de negocio para resolver el problema.

El subsistema modelo (model) será el que interactúe con la base de datos y haga de contendedor de las entidades JPA. Este subsistema recibe las peticiones del controlador cuando este necesita información almacenada en la base de datos, atenderá su consulta y posteriormente le devolverá el resultado introducido en la entidad o lista de entidades correspondiente. Además, al ser este subsistema el que se comunice con la base de datos, tendrá implementada toda la lógica necesaria para conectarse a ella y realizar cualquier operación de consulta, inserción, actualización o borrado.

7.3.5. Especificación del entorno tecnológico

Este apartado describe la tecnología, tanto software como hardware, que va a necesitar el sistema para su correcto funcionamiento.

En cuanto a hardware, es recomendable que los equipos cliente tengan los siguientes requisitos mínimos:

- Monitor SGVA.
- Ratón o dispositivo señalador compatible.
- Equipo conectado a internet.
- 512 MB de RAM.
- Intel Pentium 4 o posterior
- 350 MB de espacio libre en disco duro.

La mayoría de estos requisitos hardware para el equipo del cliente vienen dados por el navegador web recomendado a usar (Chrome o Firefox).

Por último, para el equipo servidor donde se almacenará la aplicación se requerirá un equipo con al menos las siguientes características:

- 32 GB memoria RAM.
- 10 TB de almacenamiento máximo.
- Dos Procesadores Intel Xeon 8C X7560 con 8 núcleos cada uno.

Adentrándonos en la tecnología software nuestra aplicación necesitará que el servidor de despliegue y los equipos de desarrollo utilizados sean compatibles con el estándar J2EE. El SGBD a usar será MySQL, extensible a ORACLE si en futuras versiones aumenta la complejidad de las operaciones con la base de datos.

La codificación se realizará en lenguaje JAVA usando como IDE eclipse. La JDK a usar será la 1.7. Las páginas con contenido dinámico usarán JSP, mientras las que sean estáticas se usará HTML.

La documentación generada durante el proyecto se realizará usando la suite ofimática Microsoft Office 2013.

Todas las funcionalidades de la aplicación deberán funcionar correctamente en los navegadores web Chrome y Firefox.

7.3.6. Especificación de requisitos de seguridad y operación

Estos requisitos ya fueron especificados en el análisis en los puntos 6.4.2.4. *Requisitos de seguridad* y 6.4.2.8. *Requisitos de operación* respectivamente.

Las bases y razones de los requisitos que se han recogido de estas categorías son no comprometer el correcto funcionamiento del sistema y garantizar el cumplimiento de los niveles de servicios que exigirá este en cuanto a la gestión de operaciones.

7.3.7. Estudio de la seguridad requerida en el proceso del sistema de información

En este punto se especifican las medidas de seguridad que se van a tomar con los productos obtenidos en el proceso de desarrollo del sistema.

El acceso a los documentos, durante el desarrollo, estará restringido únicamente al Project manager y al tutor del proyecto (cliente).

La disponibilidad de los documentos será total, al estar la memoria y el código alojados en un repositorio online (*jazzhub*). Este repositorio online tendrá control de versiones para minimizar los riesgos en caso de perder documentos en dispositivos locales. En la siguiente imagen se puede ver parte del log del tratamiento de las versiones del proyecto.



Ilustración 67 Control de versiones

Cada entrega que se haga al cliente del proyecto hará a este responsable de las acciones que haga con él.

7.3.8. Análisis de los riesgos del entorno tecnológico

En el caso de este proyecto estos son mínimos y casi inexistente a causa del software utilizado para su desarrollo.

7.4. Diseño de la arquitectura de soporte

En este apartado se partirá de lo especificado anteriormente en el 7.3. Definición de la arquitectura del sistema.

En esta tarea se determinan además los mecanismos genéricos de diseño que servirán posteriormente de guía para el diseño detallado del sistema de información.

7.4.1. Diseño de subsistemas de soporte

Este apartado presenta los elementos más significativos del sistema que componen su arquitectura software, como son las clases y los componentes.

El sistema a desarrollar tiene una arquitectura compuesta por tres componentes principales que son modelo, vista y controlador.

Se partirá de estos tres componentes para definir sus clases, subcomponentes, interfaces y atributos asociados. Con el objetivo de llegar al diseño detallado del sistema.

A continuación se definirán las características de estos tres componentes que confirman el sistema. Cada uno vendrá representado por los siguientes atributos:

- **Identificador:** Único para cada componente. Seguirá la regla de nombrado CO-XX, donde CO es componente y XX el número de este.
- **Tipo:** Para este caso todos serán de tipo componente.
- **Propósito:** Fin del componente.
- **Función:** Responsabilidades del componente.
- **Subordinados:** Identificación de herencias entre componentes.
- **Dependencias:** Relaciones entre los componentes.
- **Interfaz:** Servicios proporcionados por el componente.
- **Recursos:** Elementos hardware necesarios por el componente para realizar su función.
- **Referencias:** Documentación necesaria para entender el componente.
- **Proceso:** Algoritmos especiales de procesamiento.
- **Datos:** Datos internos del elemento, atributos, relaciones de interés que no se hayan indicado en la descripción, posibles valores de alguno de los elementos...

CO-00			
Nombre	Capa vista	Tipo	Componente

Propósito	Dar soporte a los usuarios con los elementos visuales de la aplicación web		
Función	Interactuar con el usuario y mostrar el resultado de sus peticiones		
Subordinados	Ninguno	Dependencias	Capa controlador y modelo
Interfaz	Ninguno	Recursos	Ninguno
Referencias	Ninguno	Proceso	
Datos	Ninguno		

Tabla 159 Componente vista

CO-01			
Nombre	Capa controlador	Tipo	Componente
Propósito	Gestiona las peticiones realizadas por el usuario		
Función	Recoge la petición del usuario y se apoya en el modelo para obtener los datos necesarios para resolverla. Devuelve el resultado a la vista para que se lo muestre al usuario		
Subordinados	Ninguno	Dependencias	Capa modelo
Interfaz	Ninguno	Recursos	Ninguno
Referencias	Ninguno	Proceso	
Datos	Ninguno		

Tabla 160 Componente controlador

CO-02			
Nombre	Capa modelo	Tipo	Componente
Propósito	Gestiona los datos de la aplicación		
Función	Gestiona las entidades de la aplicación y contiene la funcionalidad para comunicarse con la base de datos		
Subordinados	Ninguno	Dependencias	Capa controlador
Interfaz	Ninguno	Recursos	Ninguno
Referencias	Ninguno	Proceso	
Datos	Ninguno		

Tabla 161 Componente modelo

7.5. Diseño de caso de uso reales

El propósito de esta actividad es especificar el comportamiento del sistema de información para cada caso de uso, mediante objetos o subsistemas de diseño que interactúan, y determinar las operaciones de las clases e interfaces de los distintos subsistemas de diseño.

Para ello, una vez identificadas las clases participantes dentro de un caso de uso, será necesario completar los escenarios que se recogen del análisis, incluyendo las clases de diseño participantes y teniendo en cuenta las restricciones del entorno tecnológico, es decir, con detalles relacionados con la implementación del sistema.

7.5.1. Identificación de clases asociadas a un caso de uso

El objetivo es identificar las clases del diseño y/o los subsistemas cuyas instancias son necesarias para llevar a cabo el flujo de sucesos de cada caso de uso.

El nombre de las clases que se mostrarán a continuación proviene del apartado del diseño 7.6.1. *Identificación de clases de diseño*, la realización de estas dos tareas se hará en paralelo, por lo que se recomienda leer ambas si no se entiende alguna parte de una de ellas.

A continuación se adjunta una tabla en la que se muestra cada caso de uso identificado con sus respectivas clases participantes y al componente que corresponderían.

Caso de uso	Clase asociada	
	Nombre	Componente
CU-00: Acceso a la aplicación web	index.jsp	Vista
	inicio.jsp	Vista
	forgotten.jsp	Vista
	recover.jsp	Vista
	recuperar.jsp	Vista
	signup.jsp	Vista
	registro.jsp	Vista
	ServletController.java	Controlador
	LoginServlet.java	Controlador
	RecoverServlet.java	Controlador
	RegisterServlet.java	Controlador
	ActivateServlet.java	Controlador
CU-01: Cierre de sesión	User.java	Modelo
	Project.java	Modelo
	Funciones.java	Modelo
	top.jsp	Vista
	index.jsp	Vista
	ServletController.java	Controlador
CU-02: Gestión de cuenta	LogoutServlet.java	Controlador
	User.java	Modelo
	Funciones.java	Modelo
	top.jsp	Vista
	account.jsp	Vista
CU-03: Subir caso de simulación	ServletController.java	Controlador
	AccountServlet.java	Controlador
	User.java	Modelo
	upload.jsp	Vista
	UploadServlet.java	Controlador
	Project.java	Modelo
	Funciones.java	Modelo
	Simulate.java	Modelo

CU-04: Gestión de casos	inicio.jsp see.jsp shared.jsp ServletController.java SeeServlet.java RunServlet.java ShareServlet.java DownloadServlet.java DeleteServlet.java Project.java DeletedProject.java Sharing.java Simulate.java Funciones.java	Vista Vista Vista Controlador Controlador Controlador Controlador Controlador Controlador Controlador Modelo Modelo Modelo Modelo Modelo
CU-05: Gestión de permisos de caso	shared.jsp ServletController.java ShareServlet.java Project.java User.java Sharing.java	Vista Controlador Controlador Modelo Modelo Modelo
CU-06: Elegir idioma	language.jsp ServletController.java LanguageServlet.java	Vista Controlador Controlador
CU-07: Navegación entre casos de simulación	inicio.jsp ServletController.java PageServlet.java Project.java	Vista Controlador Controlador Modelo

Tabla 162 Clases asociadas a casos de uso

7.5.2. Diseño de la realización de los casos de uso

El objetivo de esta tarea es describir cómo interactúan entre sí los objetos identificados en la tarea anterior para realizar cada uno de los casos de uso del sistema de información, desde un punto de vista técnico.

Para ello se partirá de los diagramas de secuencia realizado en la actividad 6.6.1. *Identificación de clases asociadas a un caso de uso* del análisis. Salvo que en este apartado serán más completos y los objetos con los que interactúan tendrán el nombre dado en el diseño.

Por simplificación de los diagramas de secuencia el *ServletController.java* únicamente se representará en el caso de uso CU-00.

7.5.2.1. CU-00: Acceso a la aplicación web

A continuación se exponen los diagramas de secuencia del caso de uso CU-00 adaptado al nombrado y diseño de las clases del sistema.

- Un usuario registrado se loguea.

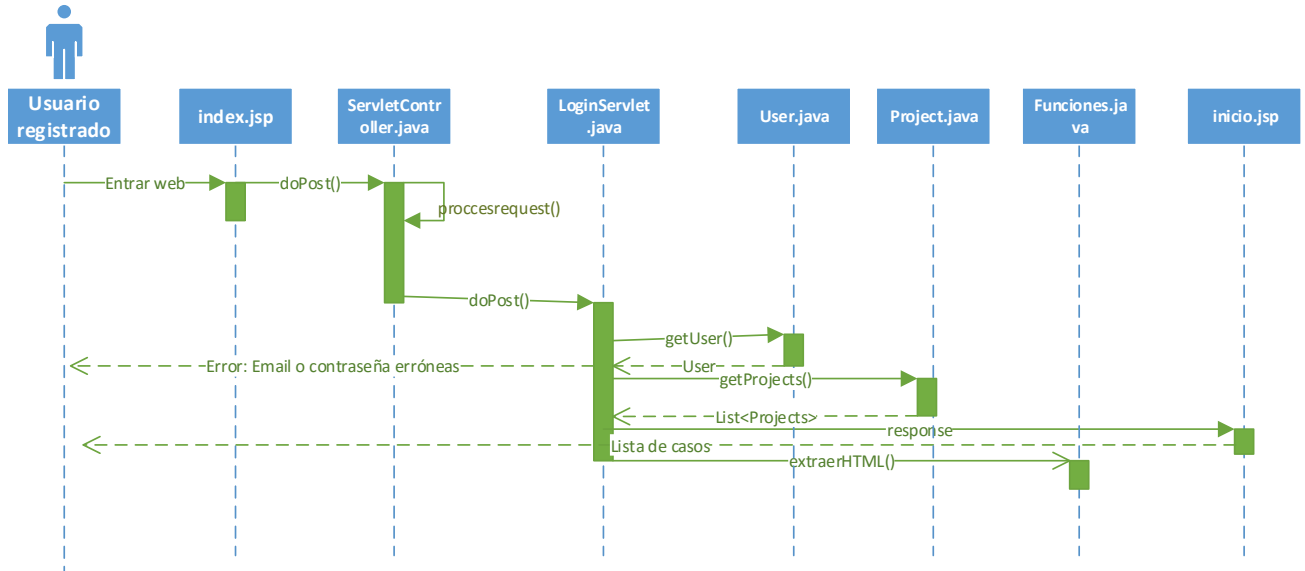
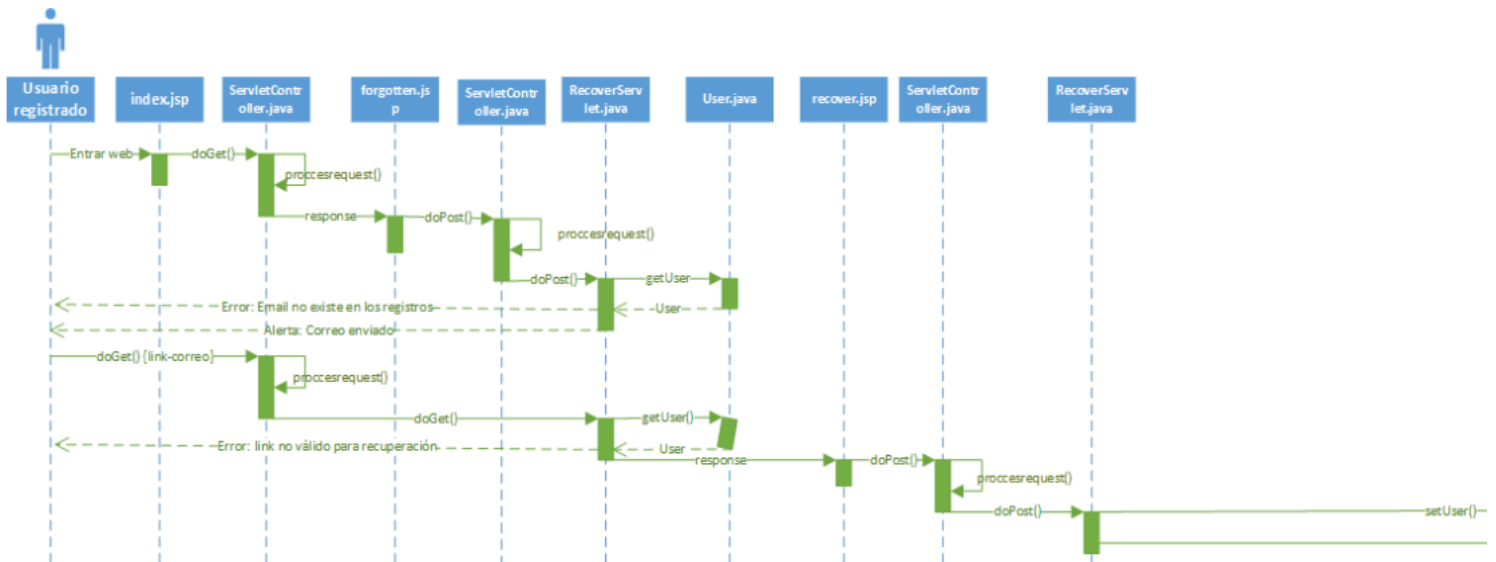


Ilustración 68 CU-00_1

- Un usuario registrado recupera la contraseña.



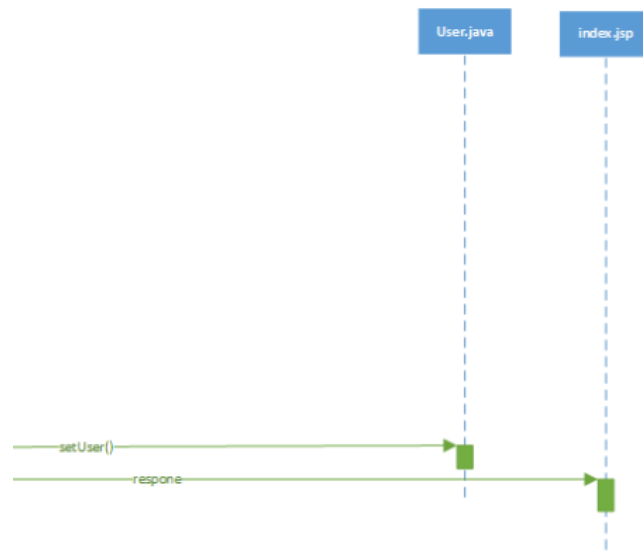


Ilustración 69 CU-00_2

- Un usuario no registrado se registra.

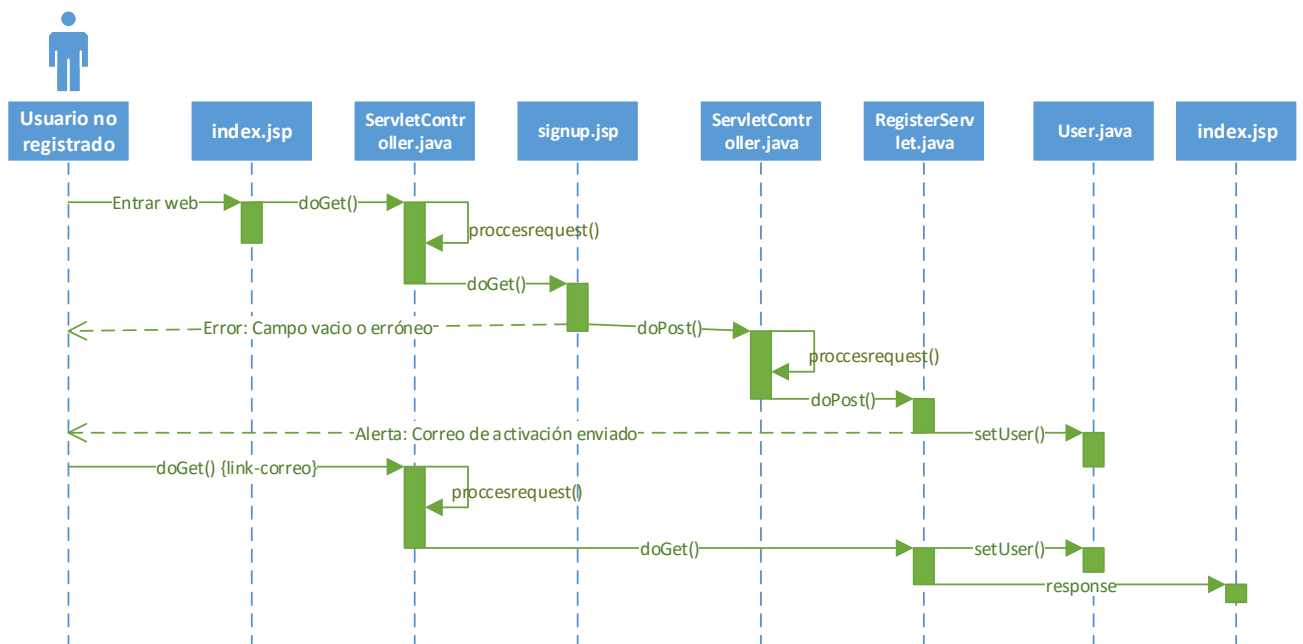


Ilustración 70 CU-00_3

7.5.2.2. CU-01: Usuario logueado cierra sesión

A continuación se exponen los diagramas de secuencia del caso de uso CU-01 adaptado al nombrado y diseño de las clases del sistema.

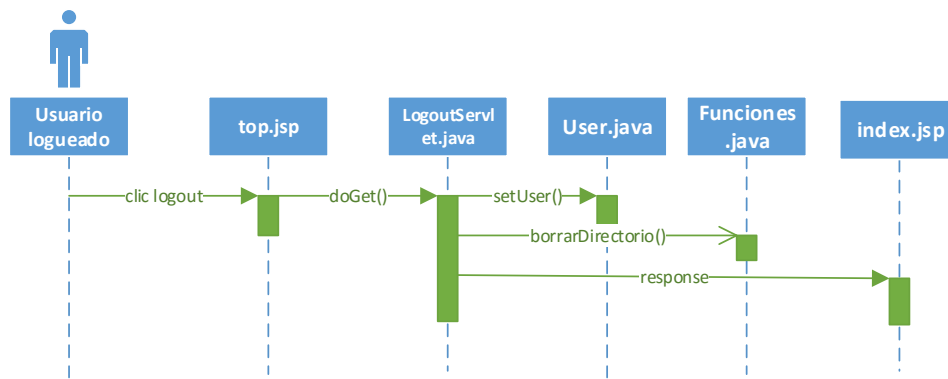


Ilustración 71 CU-01

7.5.2.3. CU-02: Gestión de cuenta

A continuación se exponen los diagramas de secuencia del caso de uso CU-02 adaptado al nombrado y diseño de las clases del sistema.

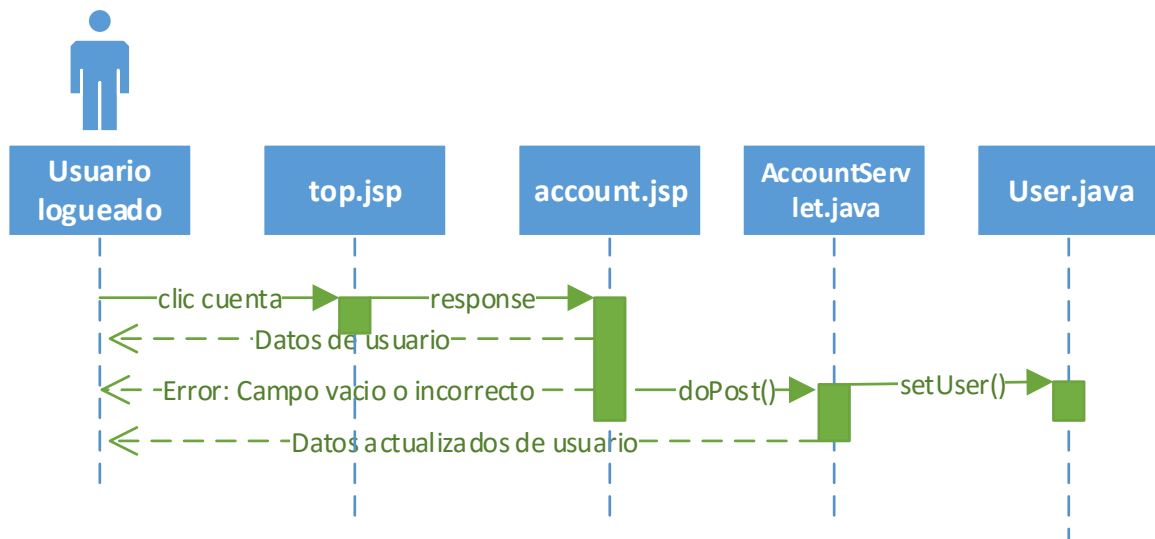


Ilustración 72 CU-02

7.5.2.4. CU-03: Subir caso de simulación

A continuación se exponen los diagramas de secuencia del caso de uso CU-03 adaptado al nombrado y diseño de las clases del sistema.

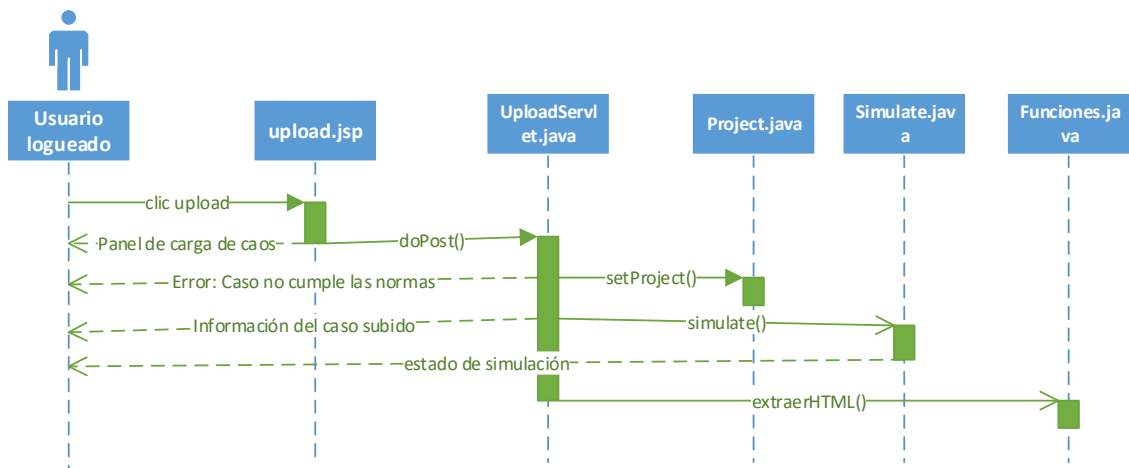


Ilustración 73 CU-03

7.5.2.5. CU-04: Gestión de casos

A continuación se exponen los diagramas de secuencia del caso de uso CU-04 adaptado al nombrado y diseño de las clases del sistema.

- *Usuario propietario o compartido visualiza la solución del caso.*

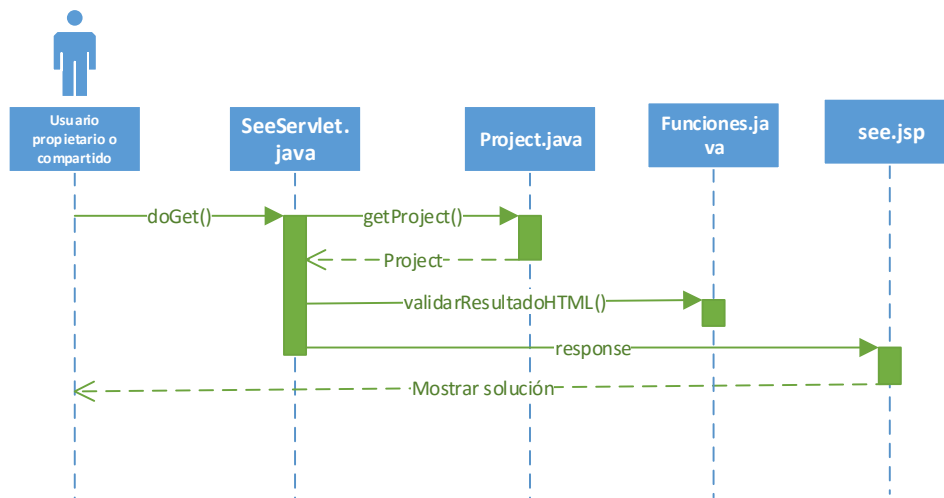


Ilustración 74 CU-04_1

- *Usuario propietario o compartido re-lanzan la simulación.*

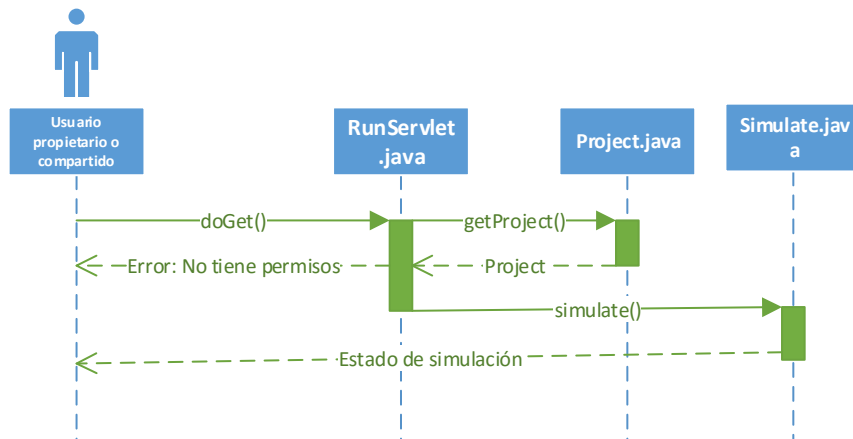


Ilustración 75 CU-04_2

- Usuario propietario o compartido comparten un caso.

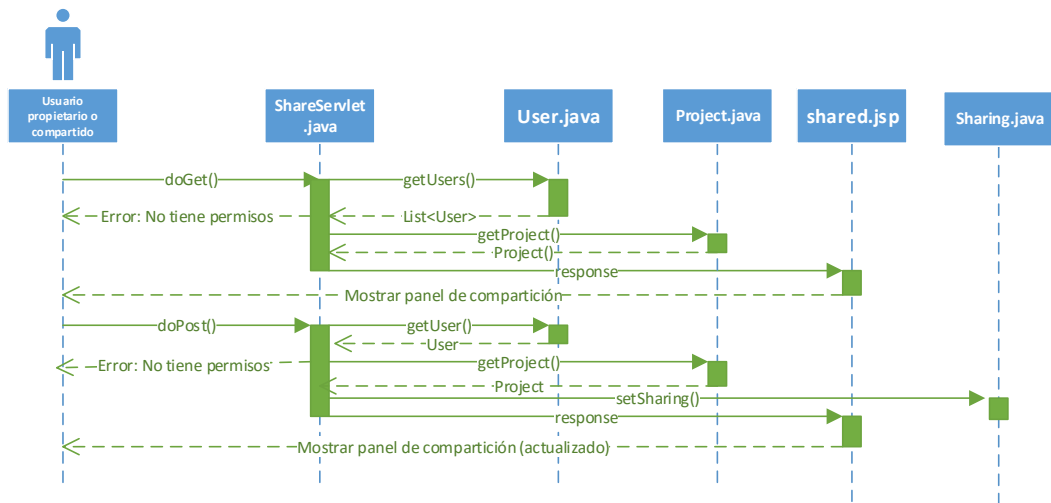


Ilustración 76 CU-04_3

- Usuario propietario o compartido descargan un caso.

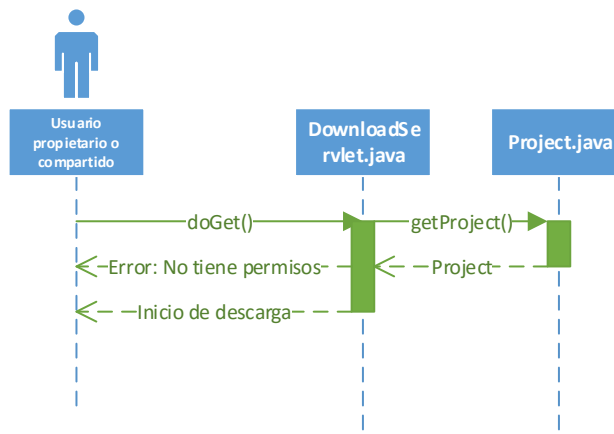


Ilustración 77 CU-04_4

- *Usuario propietario o compartido borran un caso.*

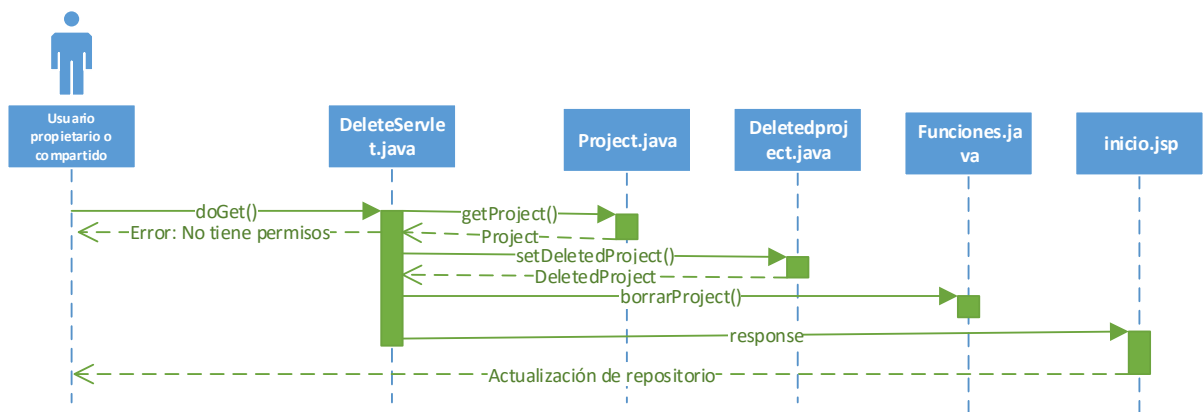


Ilustración 78 CU-04_5

7.5.2.6. CU-05: Gestión de permisos de caso

A continuación se exponen los diagramas de secuencia del caso de uso CU-05 adaptado al nombrado y diseño de las clases del sistema.

- *Usuario propietario o compartido añaden un usuario a la compartición.*

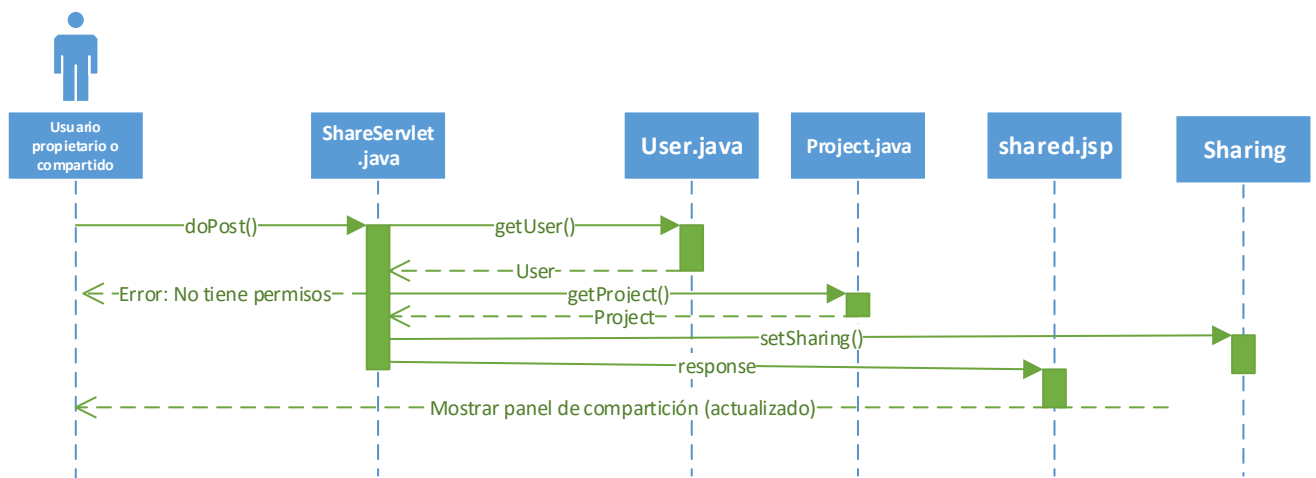


Ilustración 79 CU-05_1

- *Usuario propietario o compartido eliminan un usuario de la compartición.*

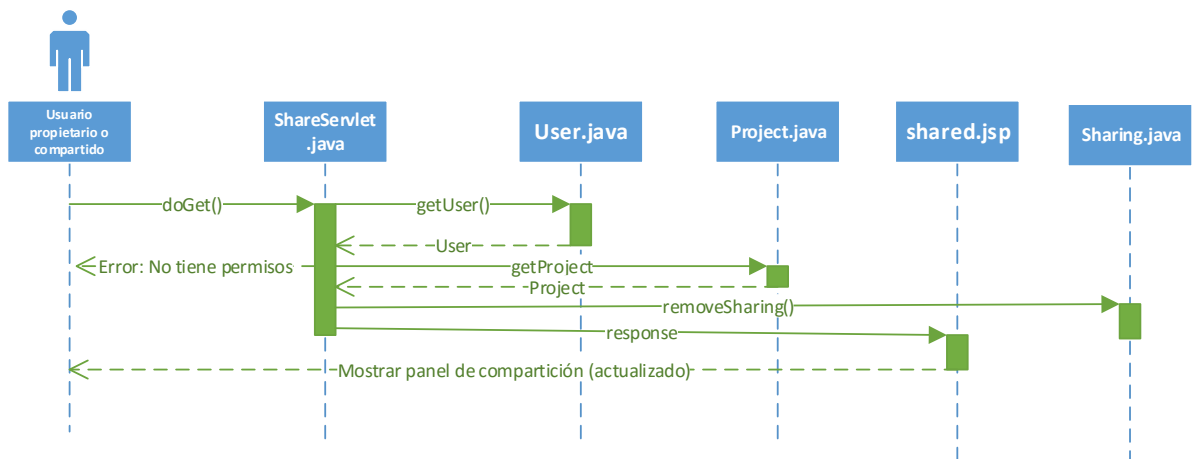


Ilustración 80 CU-05_2

- *Usuario propietario o compartido modifican permisos a un usuario.*

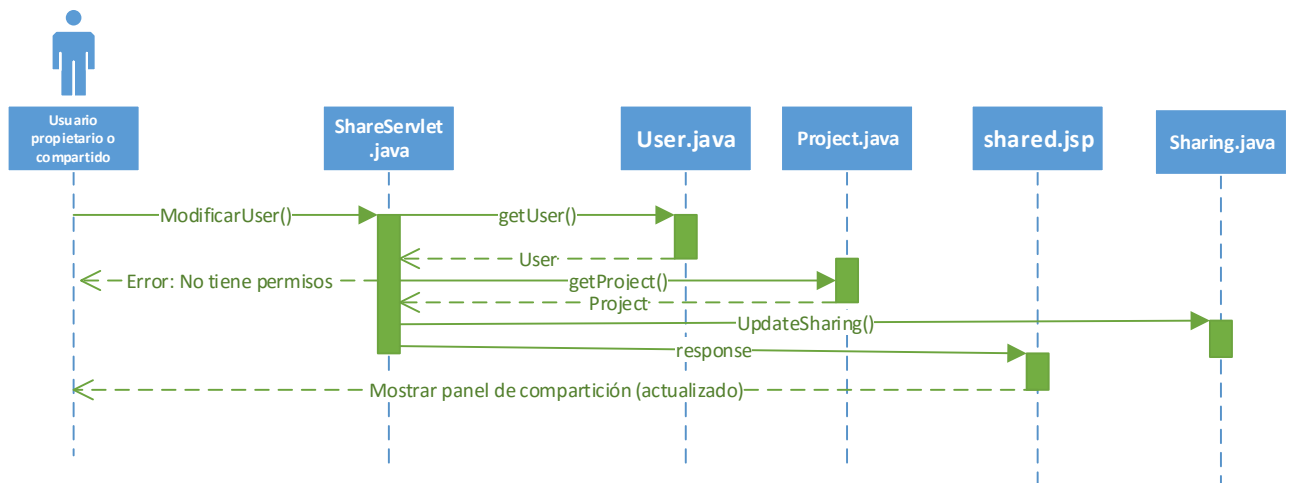


Ilustración 81 CU-05_3

7.5.2.7. CU-06: Elegir idioma

A continuación se exponen los diagramas de secuencia del caso de uso CU-06 adaptado al nombrado y diseño de las clases del sistema.

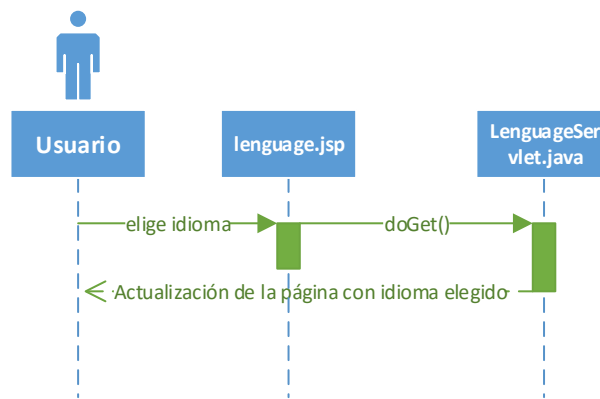


Ilustración 82 CU-06

7.5.2.8. CU-07: Navegación entre casos de simulación

A continuación se exponen los diagramas de secuencia del caso de uso CU-07 adaptado al nombrado y diseño de las clases del sistema.

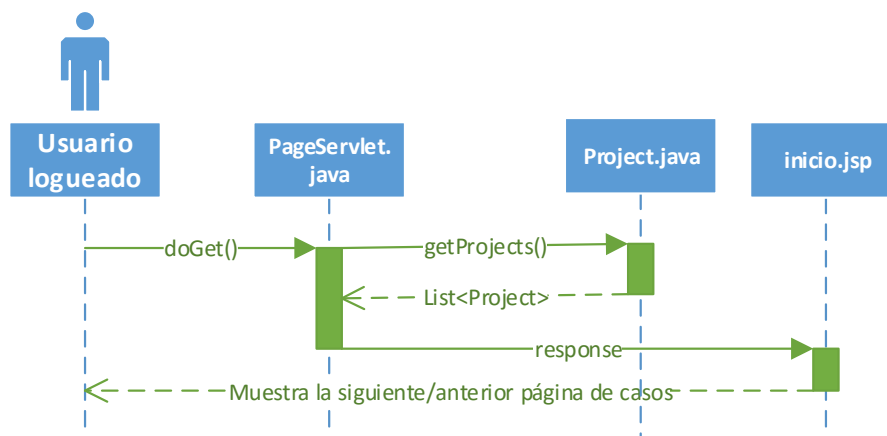


Ilustración 83 CU-07

7.5.3. Revisión de la interfaz de usuario

El objetivo de esta tarea es realizar el diseño detallado del comportamiento de la interfaz de usuario a partir de la especificación de la misma, obtenida en el proceso de análisis 6.9.3. *Especificación de formatos individuales y de la interfaz de pantalla*, y de acuerdo con el entorno tecnológico definido.

También será revisada la navegación entre las ventanas, los elementos que la forman, características, disposición y eventos relacionados con las partes que las componen.

Las imágenes que se presentan en este apartado muestran el aspecto real que tendrán las interfaces de usuario en la aplicación web.

A continuación se irá detallando cada interfaz en el mismo orden que el seguido en la etapa del análisis.

7.5.3.1. UI-00: Página de inicio

Compuesta por una caja posicionada centralmente muestra un formulario que pide al usuario el email y contraseña de registro para acceder al contenido de la aplicación.

También proporciona dos links, uno para realizar el registro y el otro para recuperar la contraseña por medio del email.

En la parte superior se mostrará parte, la selección de idioma, de la interfaz UI-01: Menú.

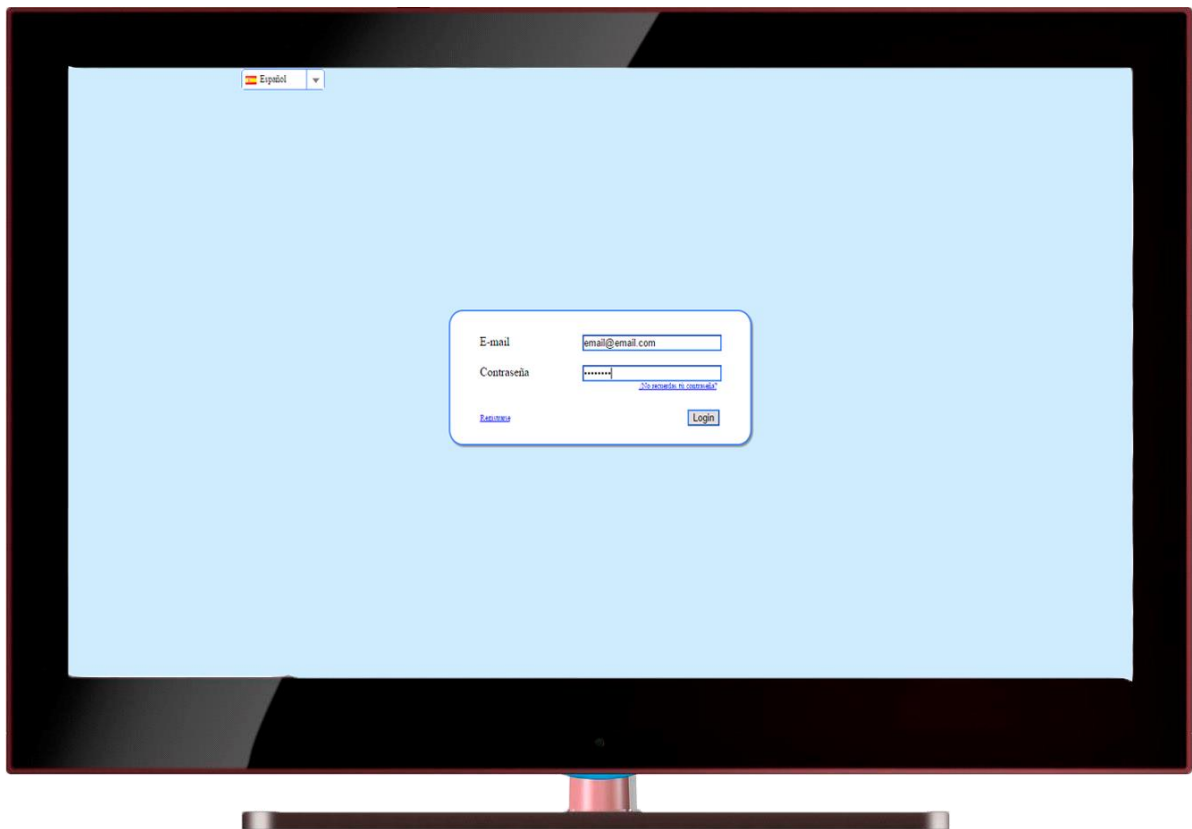


Ilustración 84 UI-00_1

La pantalla de registro muestra un formulario en el que se le pedirá al usuario su nombre y apellidos, un email y una contraseña.

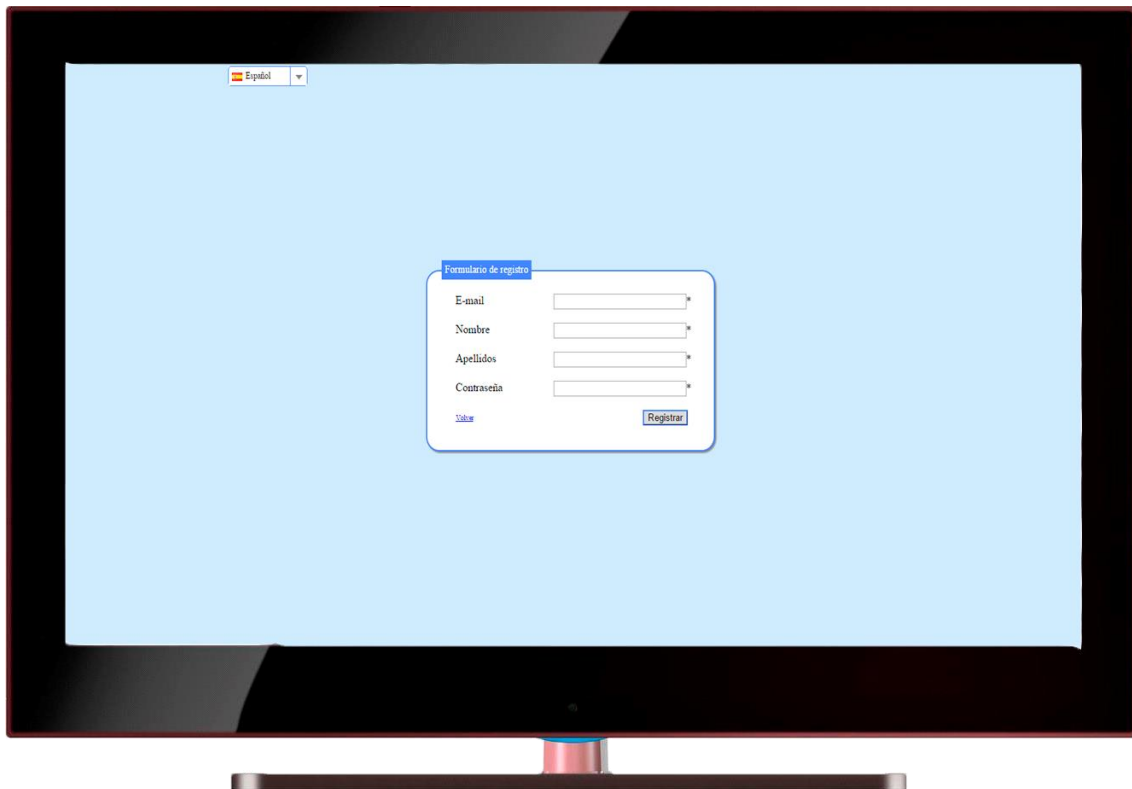


Ilustración 85 UI-00_2

La pantalla de recuperación de cuenta pedirá al usuario el email con el que se registró para mandarle un correo con los pasos para recuperar la contraseña.

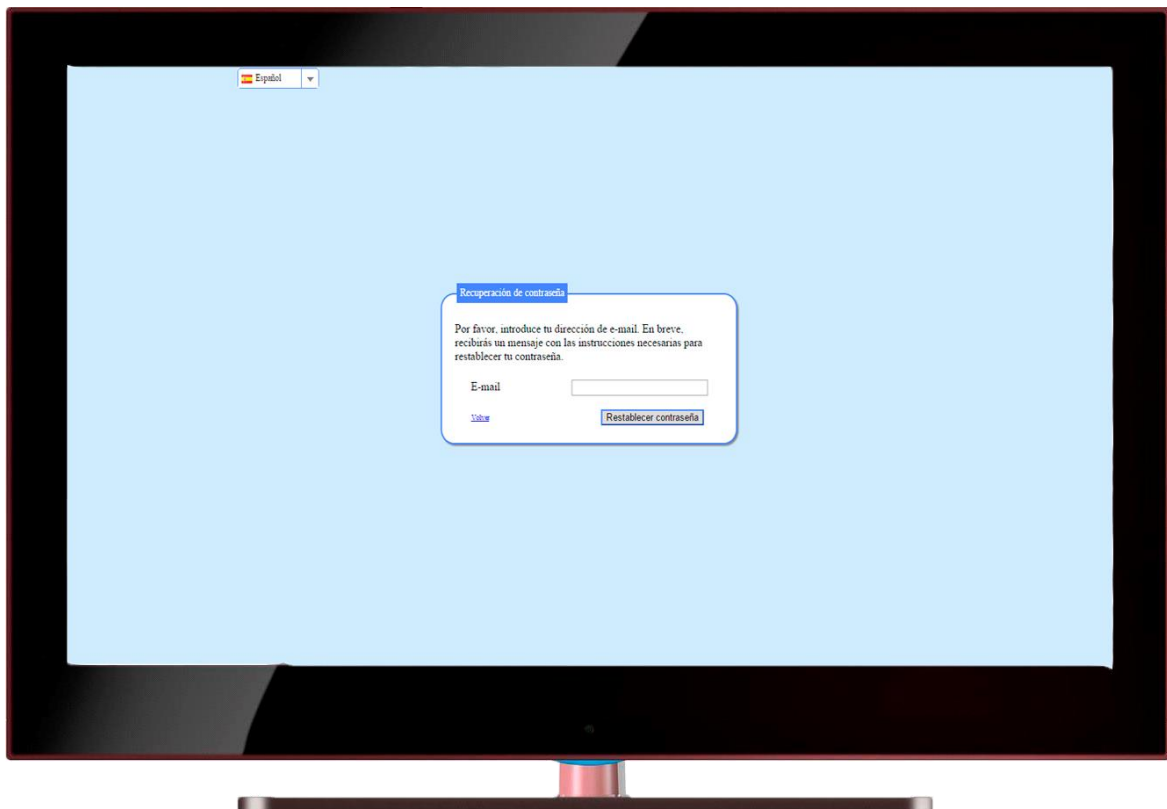


Ilustración 86 UI-00_3

ID	UI-00	Nombre	Página de inicio
Descripción	Conjunto de pantallas que permiten al usuario acceder al contenido de la aplicación		
Navegabilidad	UI-02, UI-07, UI-01 (parte), UI-07		
Elementos	Formularios, links y textos		

Tabla 163 UI-00

7.5.3.2. UI-01: Menú

Panel de navegación con el que el usuario podrá, entre otras cosas, cambiar el idioma de la aplicación, consultar el estado del servidor de simulaciones, acceder al panel de su cuenta y desconectarse.

La selección de idioma será mediante un selector, el estado del servidor usará una imagen con texto flotante, mientras que el acceso al panel de cuenta y el logout usará links.

Este panel estará integrado en todas las interfaces, en las que se requiera loguearse para acceder estará completo, mientras que en el resto sólo mostrará el selector de idioma.

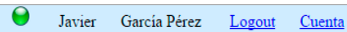


Ilustración 88 UI-01_1

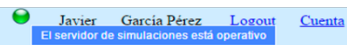
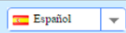


Ilustración 87 UI-01_2

ID	UI-01	Nombre	Menú
Descripción	Panel de navegación con una serie de funcionalidades secundarias para el usuario		
Navegabilidad	UI-00, UI-03		
Elementos	Input select, imagen, texto y links		

Tabla 164 UI-01

7.5.3.3. UI-02: Repositorio

Pantalla que muestra en una lista los casos que el usuario tiene subidos y con compartición en el repositorio.

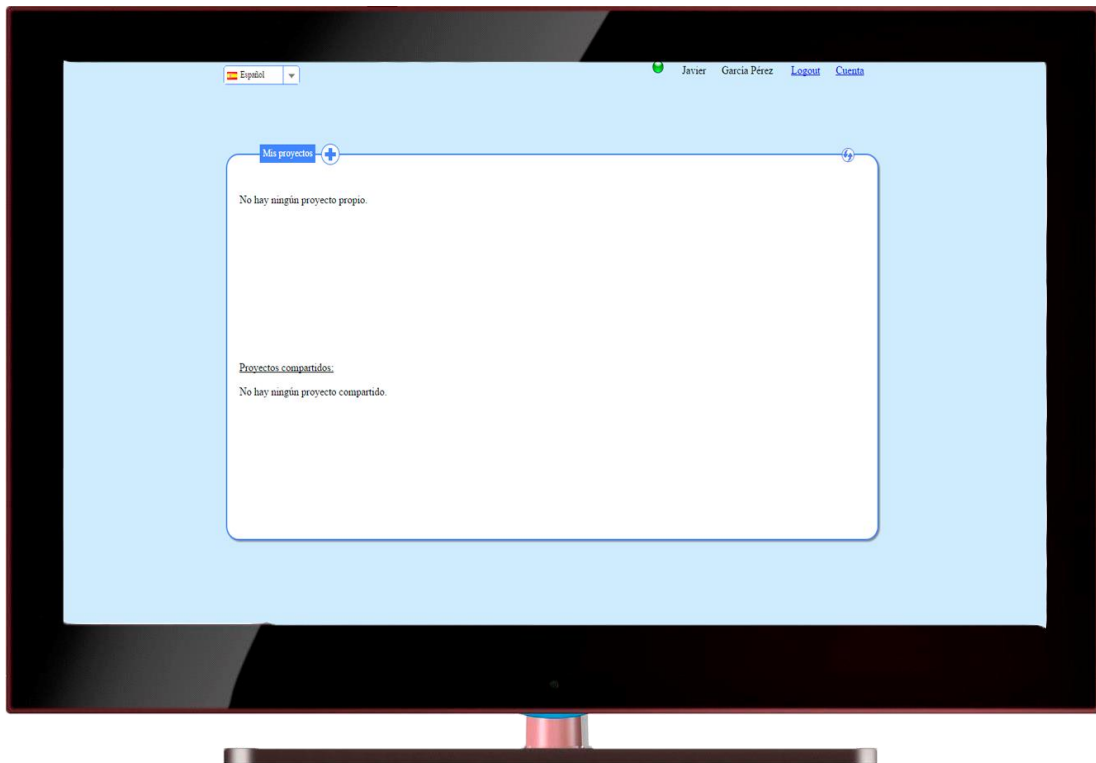


Ilustración 89 UI-02_1

Por cada caso se muestra su nombre, estado, fecha de última modificación y unos iconos con texto flotante para realizar las funcionalidades disponibles sobre él.

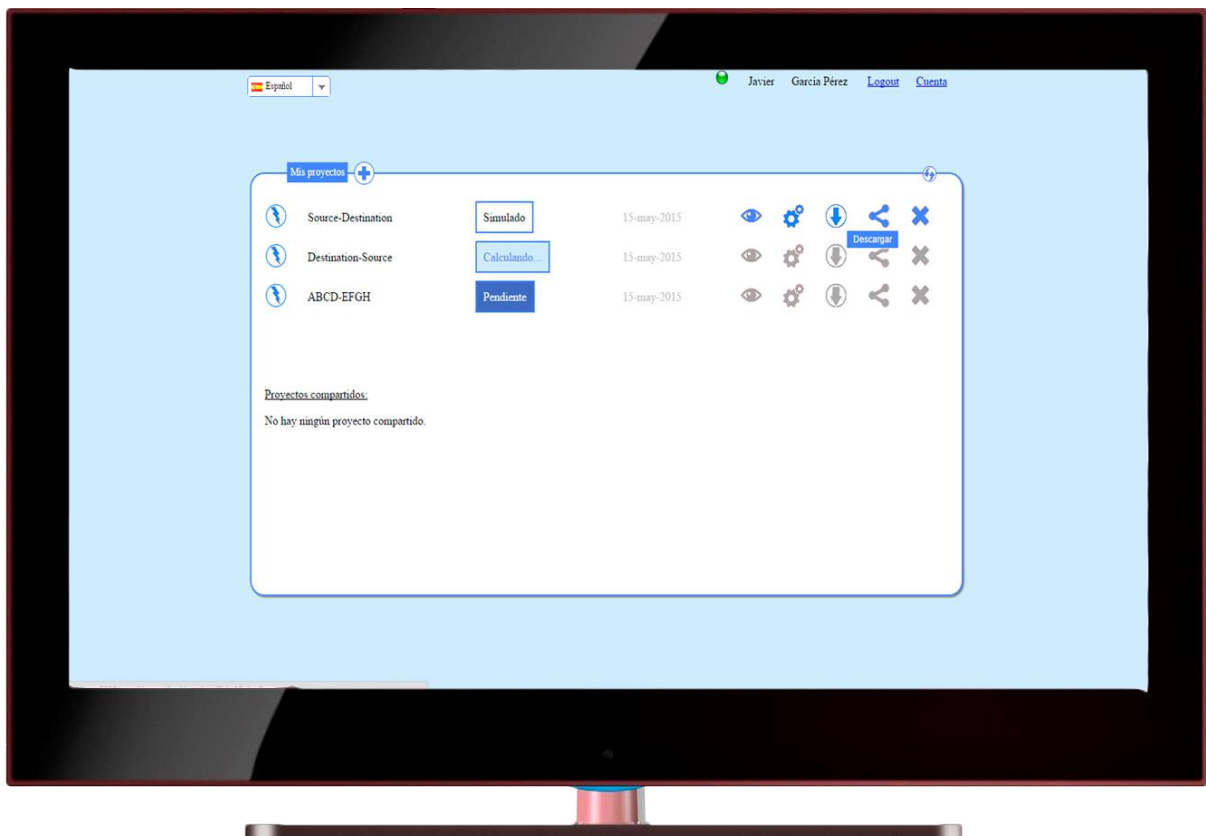


Ilustración 90 UI-02_2

En la parte superior de la caja aparecerá un link en forma de icono con una cruz que permitirá subir nuevos casos de simulación, además habrá otro para poder actualizar el repositorio y ver los cambios a tiempo real.

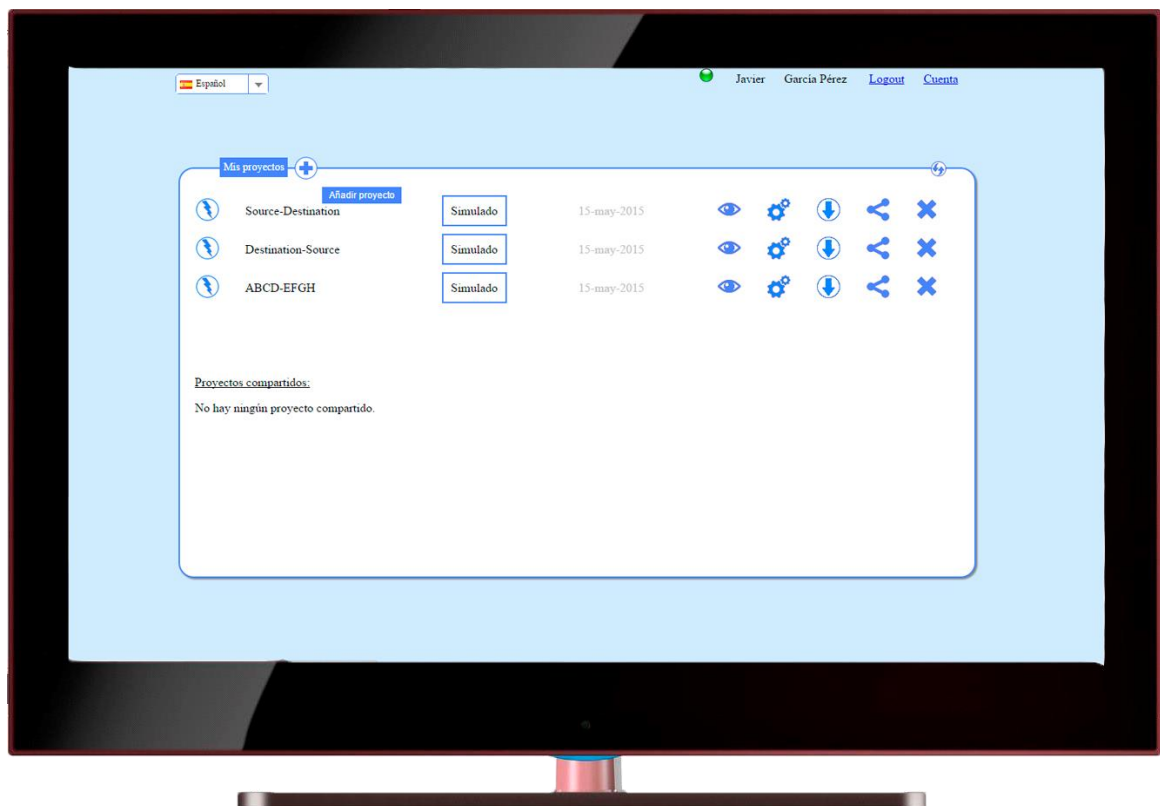


Ilustración 91 UI-02_3

En caso de haber más de cuatro proyectos propios o compartidos la interfaz habilitará la opción de página siguiente.

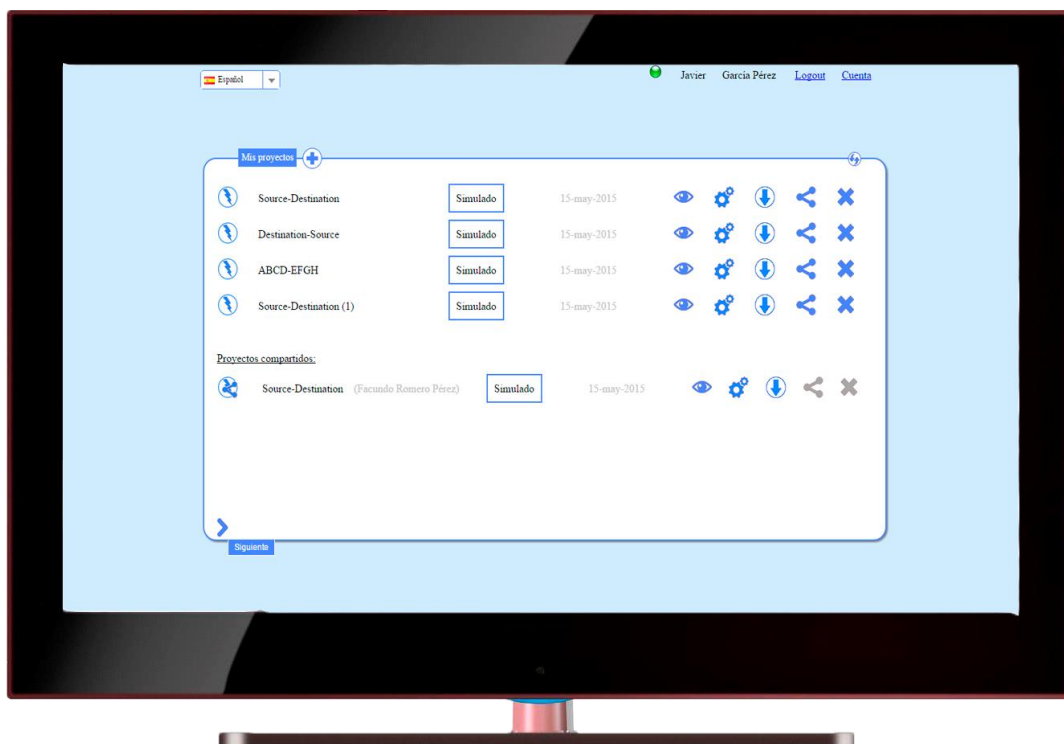


Ilustración 92 UI-02_4

ID	UI-02	Nombre	Repositorio
Descripción	Pantalla principal que mostrará el listado de casos de simulación a los que tiene acceso el usuario		
Navegabilidad	UI-01. UI-04, UI-05, UI-06		
Elementos	Lista, imágenes, links y texto		

Tabla 165 UI-02

7.5.3.4. UI-03: Gestión de cuenta

Pantalla donde el usuario podrá cambiar sus datos de registro, salvo email, mediante un formulario. En caso de cambiar la contraseña deberá confirmarla en un campo que aparecerá nuevo.

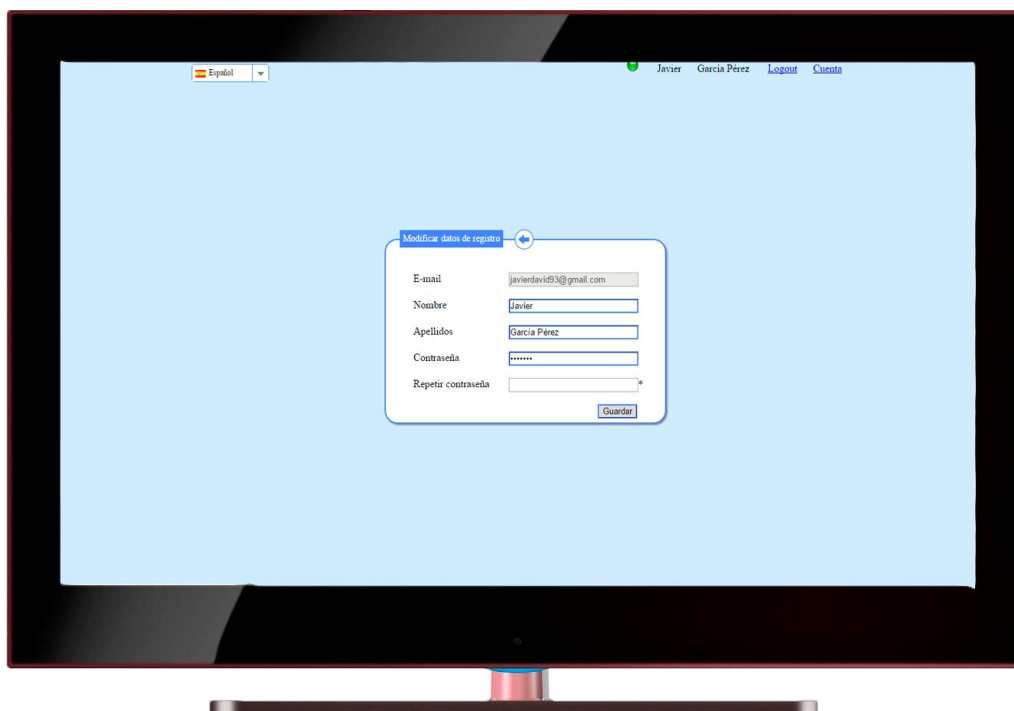


Ilustración 93 UI-03_1

Contará con un botón en la parte superior para volver atrás a la pantalla del repositorio del usuario.

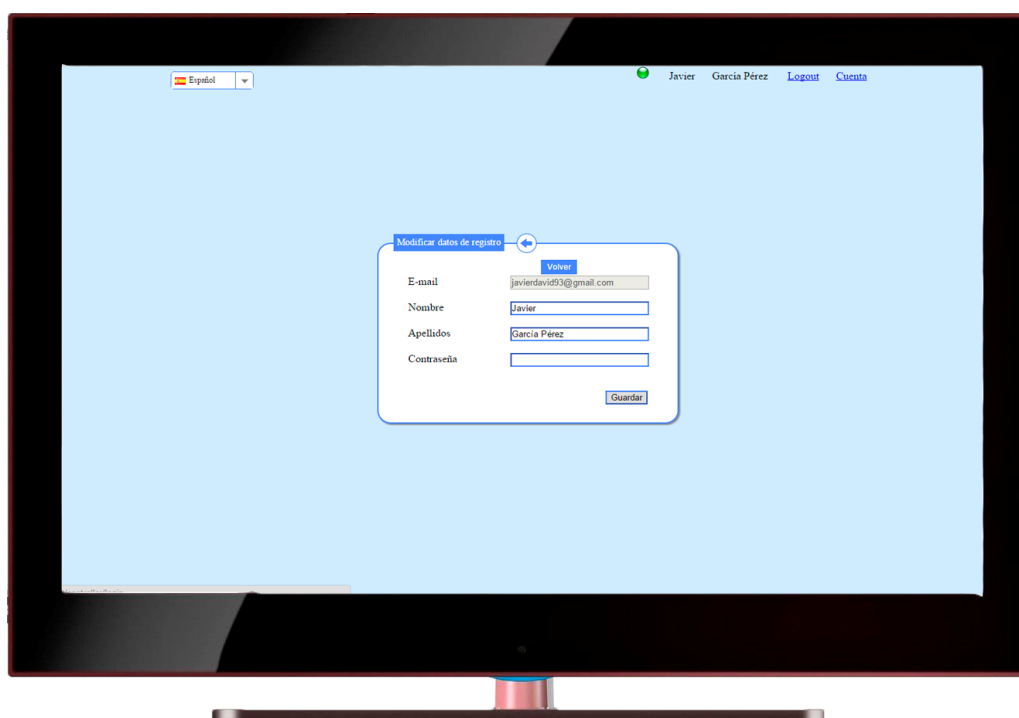


Ilustración 94 UI-03_2

ID	UI-03	Nombre	Gestión de cuenta
Descripción	Panel donde se mostrará y podrá modificar los datos de la cuenta del usuario		
Navegabilidad	UI-01. UI-03		

7.5.3.5. UI-04: Subida de casos

Pantalla donde el usuario subirá sus casos de simulación. Constará de un área donde el usuario podrá arrastrar los casos de simulación o hacer clic en ella para añadirlos manualmente. También, como la anterior interfaz, contará con un botón para volver al repositorio.

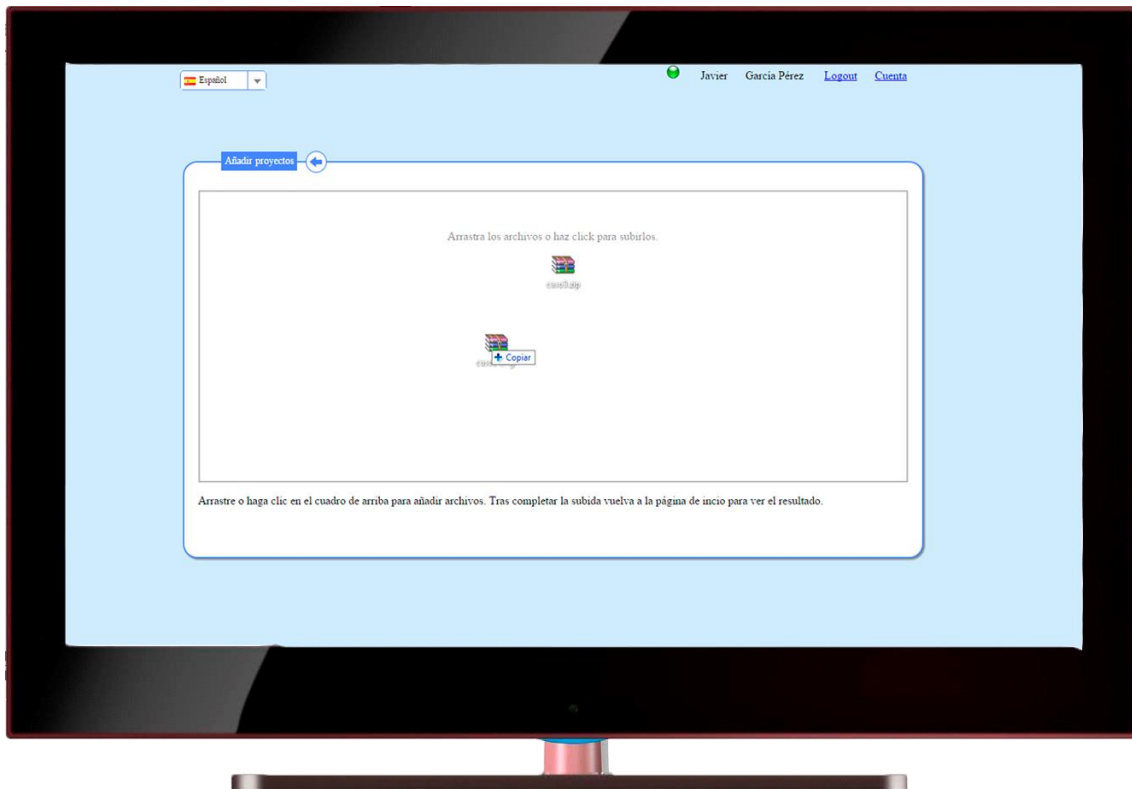


Ilustración 95 UI-04_1

Tras la subida del último caso de la cola de subidas la aplicación redirigirá automáticamente al repositorio del usuario.

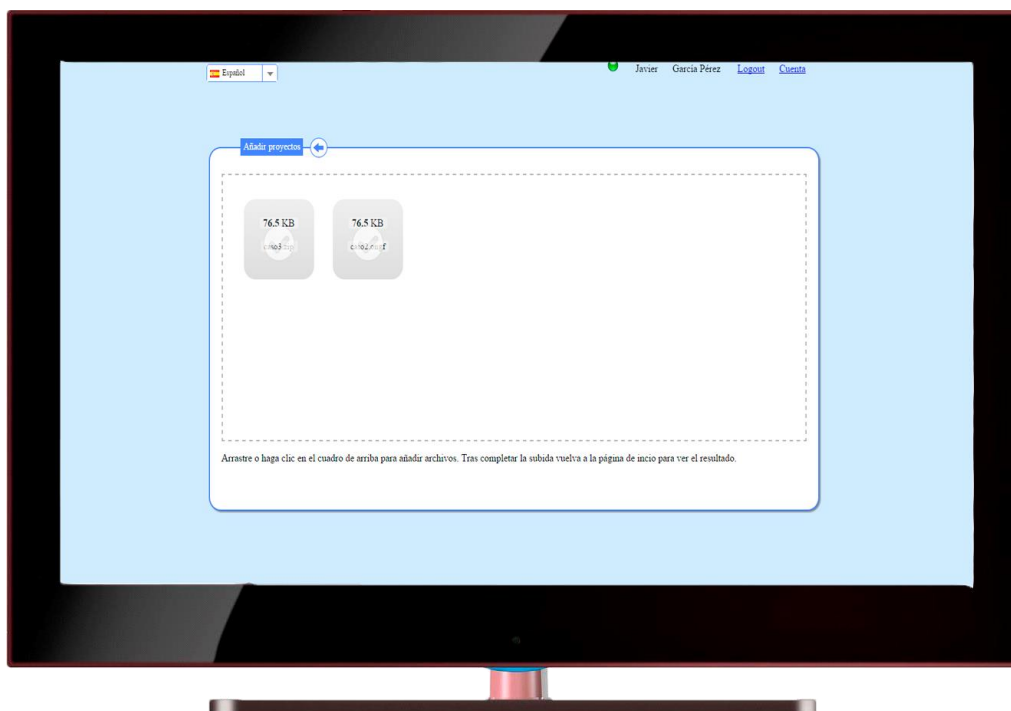


Ilustración 96 UI-04_2

ID	UI-04	Nombre	Subida de casos
Descripción	Panel de subida de casos de simulación al servidor		
Navegabilidad	UI-01. UI-02		
Elementos	Input file, texto, link		

Tabla 167 UI-04

7.5.3.6. UI-05: Panel de compartición

Ventana donde el usuario podrá crear una compartición con otro sobre un caso de simulación introduciendo su email. La interfaz contará con un autocompletar por si el usuario solo conoce el nombre o apellido del destinatario. Como las anteriores interfaces también contará con un botón para volver al repositorio en la misma posición.

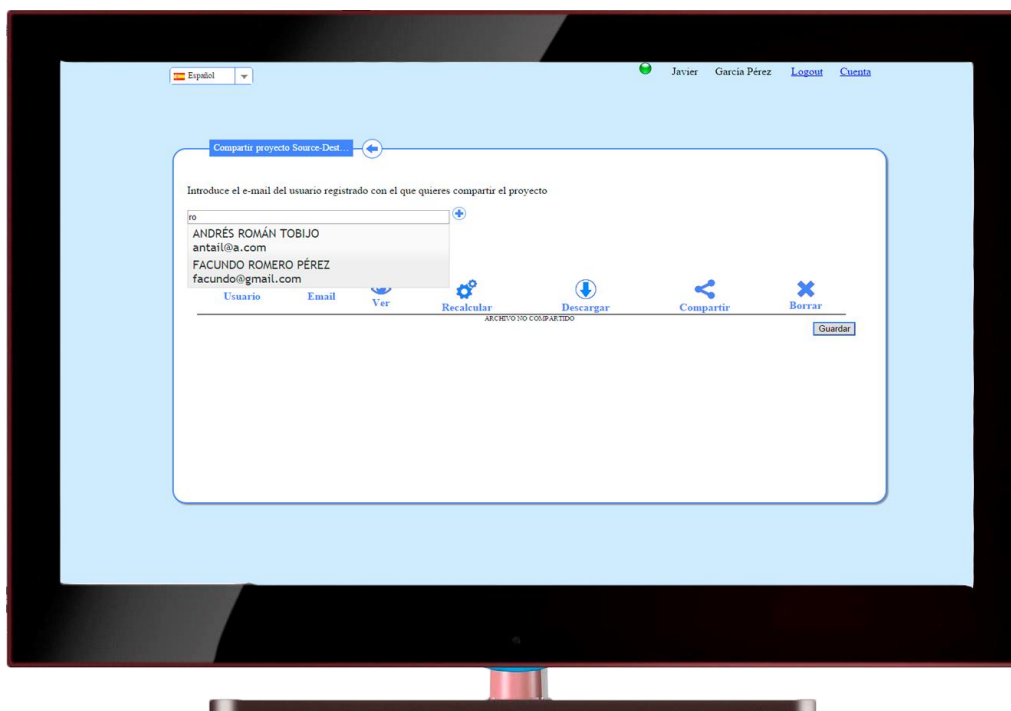


Ilustración 97 UI-05_1

Además el usuario que cree la compartición podrá seleccionar que permisos asigna a ese usuario sobre el caso.

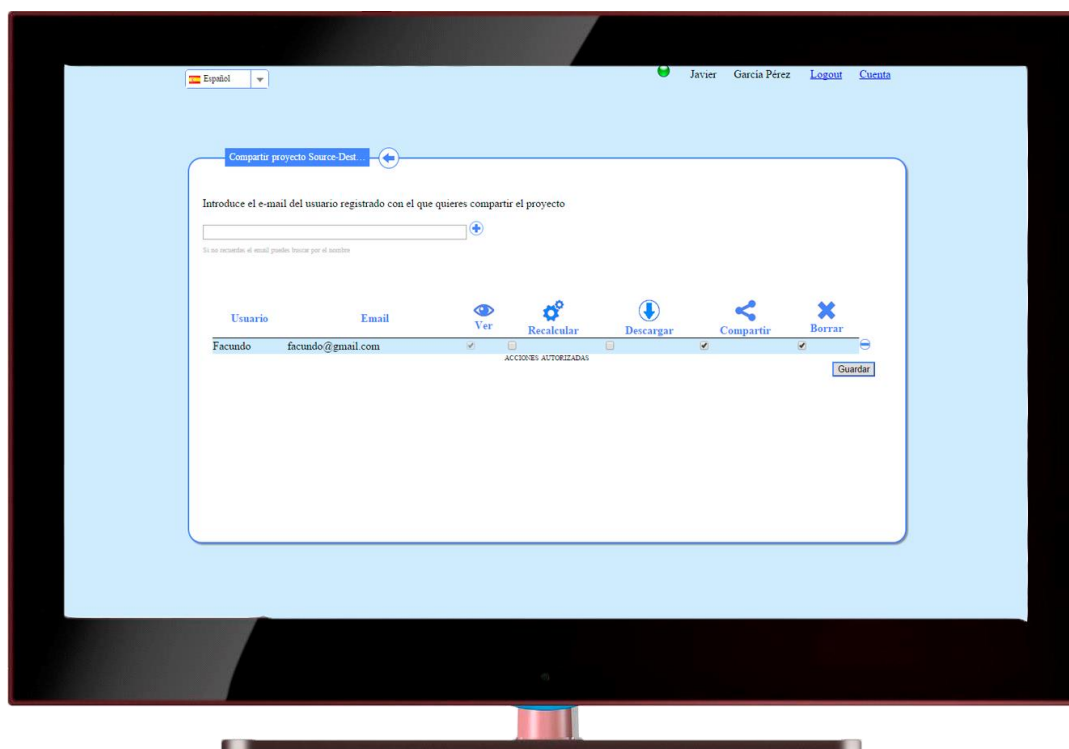


Ilustración 98 UI-05_2

ID	UI-05	Nombre	Panel de compartición
Descripción	Ventana de gestión de permisos de un caso de simulación		
Navegabilidad	UI-01. UI-02		

Elementos	Formulario, tabla, checkboxes, link y texto
-----------	---

Tabla 168 UI-05

7.5.3.7. UI-06: Vista de solución

Pantalla que mostrará la solución de la simulación de un caso. El contenido será el volcado del HTML de los archivos del caso. También contará con un botón para volver al repositorio, en la misma posición que interfaces anteriores.

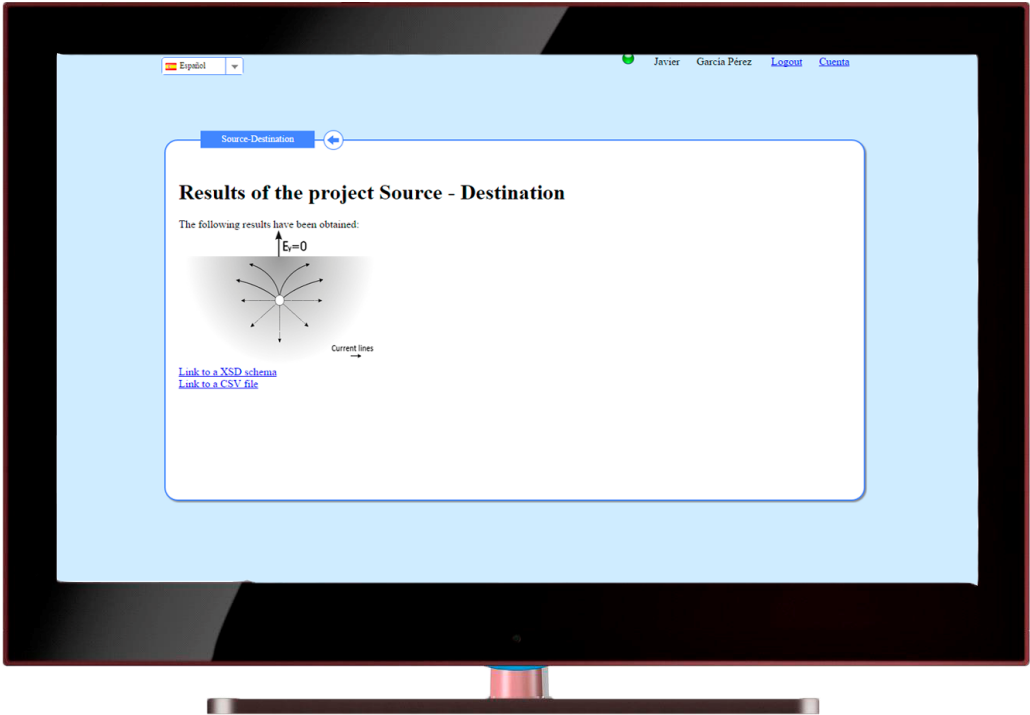


Ilustración 99 UI-06

ID	UI-06	Nombre	Vista de solución
Descripción	Ventana que mostrará la solución de un caso de simulación		
Navegabilidad	UI-01. UI-02		
Elementos	Link, texto, HTML externo		

Tabla 169 UI-06

7.5.3.8. UI-07: Error/Información

Pantalla auxiliar que mostrará mensajes al usuario tras completar una transacción o suceder un fallo.

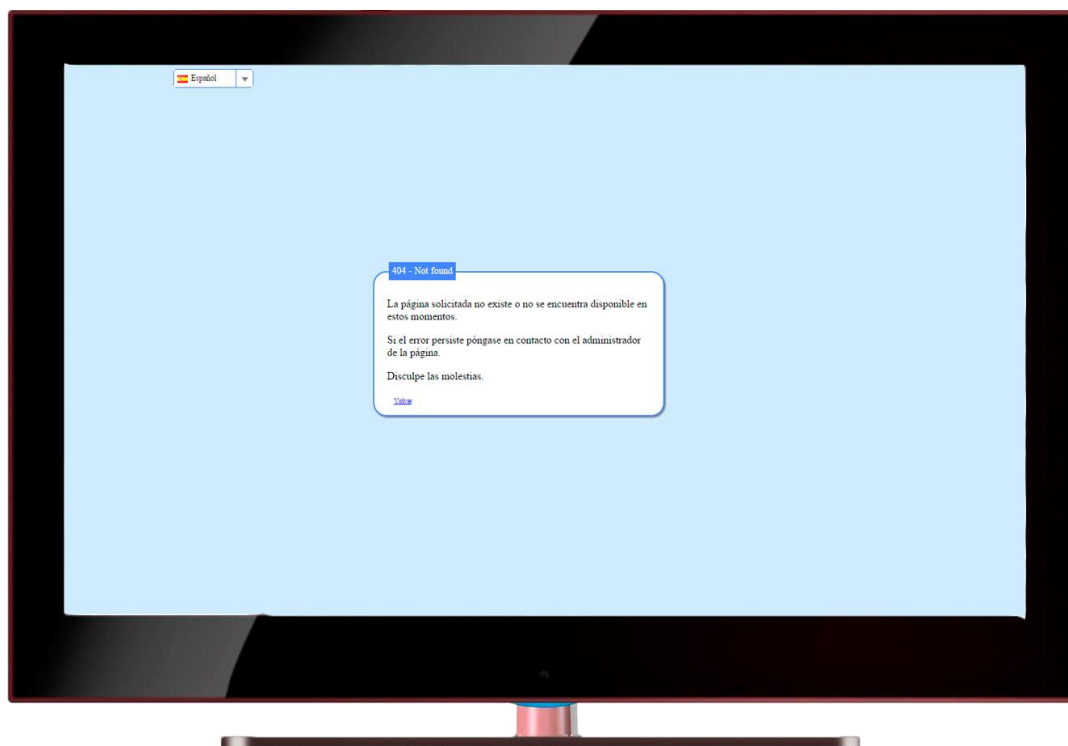


Ilustración 100 UI-07_1

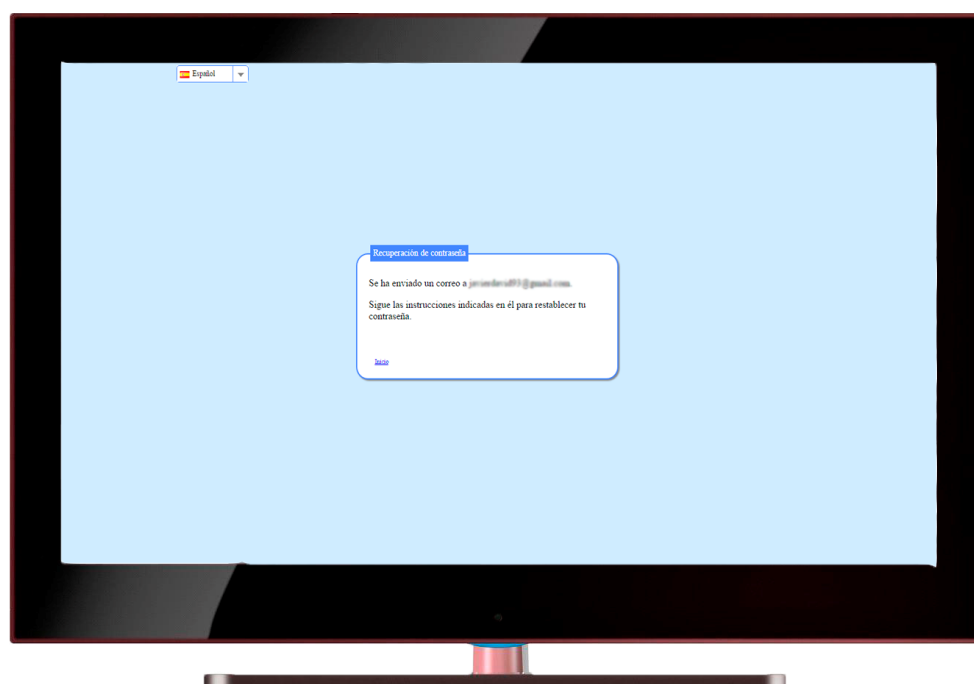


Ilustración 101 UI-07_2

ID	UI-07	Nombre	Error/Información
Descripción	Ventana de aviso de un suceso		
Navegabilidad	UI-00. UI-01 (parte), UI-02		
Elementos	Link, texto		

Tabla 170 UI-07

7.5.4. Revisión de subsistemas de diseño e interfaces

En esta tarea se describe cada caso de uso en términos de los subsistemas que van a participar y las interfaces que se requieren entre ellos.

Se expone la siguiente tabla que relacionará estos tres términos: los casos de uso, subsistemas e interfaces.

Caso de uso	Substima(s) asociado(s)	Interfaz(ces) usada(s)
CU-00	Modelo, Vista, Controlador	UI-00, UI-07
CU-01	Modelo, Vista, Controlador	UI-01
CU-02	Modelo, Vista, Controlador	UI-01, UI-03
CU-03	Modelo, Vista, Controlador	UI-02, UI-04
CU-04	Modelo, Vista, Controlador	UI-02, UI-06
CU-05	Modelo, Vista, Controlador	UI-05
CU-06	Vista, Controlador	UI-01
CU-07	Modelo, Vista, Controlador	UI-2, UI-07

Tabla 171 Revisión CU-Subsistemas-UI

7.6. Diseño de clases

Esta actividad se ha realizado en paralelo con los dos puntos anteriores de este diseño del sistema. En esta actividad se encuentra el modelo de clases identificado en el análisis, que se puede consultar en el punto 6.7. *Análisis de clases*, transformado en un modelo de clases de diseño. Este nuevo modelo recogerá de forma detallada la especificación de cada una de las clases.

7.6.1. Identificación de clases de diseño

El objetivo de esta tarea es revisar el modelo de clases obtenido en el proceso de análisis, identificando las clases que completen dicho modelo para cubrir la funcionalidad requerida en los casos de uso.

En función del patrón elegido de MVC se irán enumerando las clases. Este primer apartado será más general al tratar sólo de los nombres y funciones de las clases, en los siguientes ya se entrará en asociaciones, agregaciones, atributos... de las clases.

7.6.1.1. *Capa vista*

Las clases de esta capa son las encargadas en interactuar con el usuario, ya que envían sus peticiones al controlador y muestran el resultado de estas.

Al tratarse de archivos JSP en esta capa simplemente se hará una descripción del propósito de cada uno, será en las capas controlador y modelo donde se indague en los métodos de las clases que contienen. Los principales JSP de este subsistema son los siguientes:

- **index.jsp:** Es la página de inicio de la aplicación. Su principal función es la de proporcionar al usuario las formas de acceder al contenido principal de la aplicación.
- **signup.jsp:** Página que proporciona el formulario de registro para nuevos usuarios que quieren usar la aplicación.
- **forgotten.jsp:** Página que proporciona al usuario la opción de recuperar su contraseña partiendo de su email de registro.
- **account.jsp:** Proporciona al usuario sus datos de registro y le da la posibilidad de modificarlos a excepción del email.
- **activate.jsp:** Página auxiliar encargada de notificar al usuario que su cuenta ha sido activada cuando este visita el link del mensaje de su correo.
- **inicio.jsp:** Página que actuara como escaparate del repositorio del usuario, proporcionándole el acceso a todas las funcionalidades que puede realizar sobre los casos de simulación.
- **language.jsp:** Encargada de mostrar los idiomas disponibles al usuario y de enviar la petición de traducción cuando el idioma es cambiado.
- **recover.jsp:** Página auxiliar mostrada al usuario tras visitar el link de recuperación de contraseña que le llega al correo. En él tendrá que indicar su nueva contraseña.
- **recuperar.jsp:** Página auxiliar que notifica al usuario que un correo ha sido enviado a su email para recuperar la contraseña.
- **registro.jsp:** Mismo caso que el anterior pero para notificar que el correo que le llega es para activar su cuenta tras el registro.
- **see.jsp:** Página encargada de mostrar el resultado de la simulación de un caso al usuario.
- **shared.jsp:** Página que muestra el estado de la comparticiones sobre un caso de un usuario. También permite la gestión al usuario de los permisos sobre los usuarios con acceso compartido a ese caso.
- **top.jsp:** Página con el menú superior que incluye las opciones de ver el estado del servidor de simulaciones, acceder al panel de la cuenta y desconectarse.
- **upload.jsp:** Página que proporciona al usuario el sistema para subir casos de simulación al servidor.
- **error.jsp, error-login.jsp, error-allowed.jsp, 404.jsp, 400.jsp:** Páginas en cargadas de aparecer cuando ocurre un determinado error o fallo en el sistema.

7.6.1.2. *Capa controlador*

Las clases que engloban esta capa son las encargadas de recibir y tramitar las solicitudes del usuario.

Además en este sistema también gestionarán cierta parte de la lógica de negocio de la aplicación web.

A continuación se ira clase por clase identificando y describiendo sus métodos y la finalidad general.

- **ServletController.java**

```

ServletController
-comunicacion : Comunicacion
+init(config : servletConfig) : void
#procesRequest(request : HttpServletRequest, response : HttpServletResponse) : void
#doGet(request : HttpServletRequest, response : HttpServletResponse) : void
#doPost(request : HttpServletRequest, response : HttpServletResponse) : void

```

Ilustración 102 ServletController.java

Esta clase actúa como punto central de todas las peticiones y se encarga de llamar al servlet encargado de cada una tras procesar y ver que es correcta la petición. Las características de sus métodos son las siguientes:

- **init():** Es el encargado de hacer el llamamiento de la clase que realiza el Heartbeat por medio del objeto Comunicación. Este método se activará cada vez que se despliega la aplicación web el servidor.
- **procesRequest():** Método encargado de tramitar la petición del cliente. Redirigirá esta al servlet correspondiente si es correcta. Es llamado por *doGet()* y *doPost()*.
- **doGet() y doPost():** Reciben las peticiones del usuario y se la mandan a *procesRequest()*.

- **AccountServlet.java**

```

AccountServlet
#doPost(request : HttpServletRequest, response : HttpServletResponse) : void

```

Ilustración 103 AccountServlet.java

Este servlet será el encargado de procesar la petición de actualización de los datos de registro del usuario.

- **doPost():** Recibe el valor actualizado de los datos del usuario y llama a la clase encargado de actualizarlo en la base de datos.

- **ActivateServlet.java**

```

ActivateServlet
#doGet(request : HttpServletRequest, response : HttpServletResponse) : void

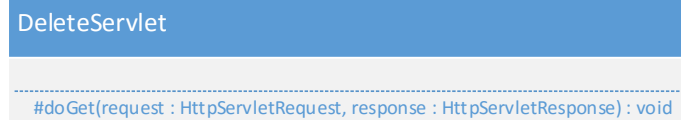
```

Ilustración 104 ActivateServlet.java

Servlet encargado de procesar la activación de un nuevo usuario registrado en la aplicación web.

- **doGet():** Recibe la petición de activar una cuenta de usuario. Comprueba que los parámetros de la petición son los correctos y llama al método encargado de actualizar en la base de datos al usuario.

- **DeleteServlet.java**



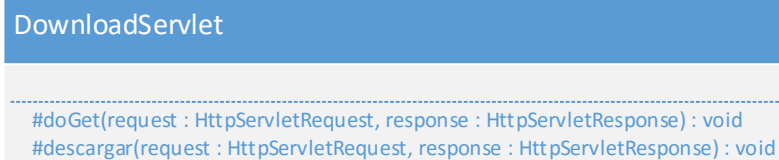
The diagram shows a class named DeleteServlet. Below the class name, there is a method signature: #doGet(request : HttpServletRequest, response : HttpServletResponse) : void. The class name is in a blue box, and the method signature is in a light blue box.

Ilustración 105 DeleteServlet.java

Este servlet se encargará de procesar las peticiones de borrado de casos de simulación.

- **doGet():** Recibe la petición de usuario de borrar un proyecto. Comprueba si el caso es de ese usuario o compartido. Aplica borrado físico del proyecto y lo elimina del repositorio de todos los que tuvieran acceso a él.

- **DownloadServlet.java**



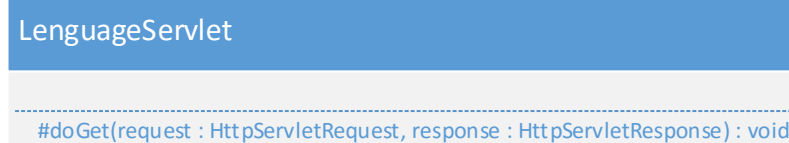
The diagram shows a class named DownloadServlet. Below the class name, there are two method signatures: #doGet(request : HttpServletRequest, response : HttpServletResponse) : void and #descargar(request : HttpServletRequest, response : HttpServletResponse) : void. The class name is in a blue box, and the method signatures are in a light blue box.

Ilustración 106 DownloadServlet.java

Servlet encargado de tramitar la descarga de un caso de simulación.

- **doGet():** Recibe la petición del usuario y busca la localización del caso en el servidor en base a lo guardado en la base de datos.
- **descargar():** Prepara y lanza la descarga del archivo del caso.

- **LanguageServlet.java**



The diagram shows a class named LanguageServlet. Below the class name, there is a method signature: #doGet(request : HttpServletRequest, response : HttpServletResponse) : void. The class name is in a blue box, and the method signature is in a light blue box.

Ilustración 107 LanguageServlet.java

Clase encargada de procesar el cambio de idioma en la aplicación web.

- **doGet():** Recibe la petición del usuario de cambio de lenguaje, la aplica y le redirige a la página donde estaba con el cambio realizado.

- **LoginServlet.java**

LoginServlet
<pre>#doGet(request : HttpServletRequest, response : HttpServletResponse) : void #doPost(request : HttpServletRequest, response : HttpServletResponse) : void</pre>

Ilustración 108 LoginServlet.java

Servlet llamado tras el login y cada vez que se actualiza el repositorio. Es el encargado de recopilar los casos del usuario y lanzar su descompresión en las carpetas temporales.

- **doGet():** Recibe las peticiones de actualización del repositorio del usuario. Lanza las funciones que buscan en la base de datos si hay cambios. También controla el número de páginas del repositorio en base a la cantidad de casos. Retorna la lista de casos del usuario.
- **doPost():** Recibe la petición del usuario tras que este realice el login. Hace la misma tarea de *doGet()* más la de lanzar la descompresión del HTML de los casos en las carpetas temporales.

- **LogoutServlet.java**

LogoutServlet
<pre>#doGet(request : HttpServletRequest, response : HttpServletResponse) : void</pre>

Ilustración 109 LogoutServlet.java

Servlet encargado de invalidar la sesión del usuario y de borrar sus carpetas temporales tras el cierre.

- **doGet():** Recibe la solicitud de cerrar sesión por parte del usuario Invalida esta y lanza la función de borrado del HTML descomprimido de sus casos.

- **RunServlet.java**

RunServlet
<pre>#doGet(request : HttpServletRequest, response : HttpServletResponse) : void</pre>

Ilustración 110 RunServlet.java

Clase encargada de lanzar la simulación de un caso.

- **doGet():** Recibe la solicitud del usuario de lanzar la simulación de un caso. Actualiza su fecha de modificación en la base de datos y lanza la simulación por medio de las funciones del objeto Simulate.

- **RegisterServlet.java**



The diagram shows a class named 'RegisterServlet' in a blue box. Below it, in a light blue box, is the method signature: `#doPost(request : HttpServletRequest, response : HttpServletResponse) : void`.

Ilustración 111 RegisterServlet.java

Servlet encargado de realizar el trámite de la creación de la cuenta de un nuevo usuario de la aplicación.

- **doPost():** Recibe la solicitud de registro con los parámetros con los valores del nuevo usuario y llama a la función encargada de guardar al usuario en la base de datos. Se encarga de crear el email de activación de cuenta que posteriormente recibirá el usuario.

- **PageServlet.java**



The diagram shows a class named 'PageServlet' in a blue box. Below it, in a light blue box, is the method signature: `#doGet(request : HttpServletRequest, response : HttpServletResponse) : void`.

Ilustración 112 PageServlet.java

Se encargará de realizar el proceso de siguiente y anterior de página en el repositorio y de obtener los casos de simulación de la nueva página.

- **doGet():** Recibe la petición del usuario de cambio de página. Ve si es la siguiente o anterior y según la elección busca los casos a mostrar.

- **RecoverServlet.java**



The diagram shows a class named 'RecoverServlet' in a blue box. Below it, in a light blue box, are two method signatures: `#doGet(request : HttpServletRequest, response : HttpServletResponse) : void` and `#doPost(request : HttpServletRequest, response : HttpServletResponse) : void`.

Ilustración 113 RecoverServlet.java

Encargado de realizar el proceso de recuperación de contraseña.

- **doGet():** Recibe la petición del usuario tras recibir este el link de recuperación a su email. Comprueba el link es válido.
- **doPost():** Recibe el email de registro del usuario que quiere recuperar la cuenta y manda el email al usuario si este existe. También es el encargado de recibir la nueva contraseña y actualizarla en la base de datos.

- **SeeServlet.java**

```

SeeServlet
-TEMP_DIR : String
#doGet(request : HttpServletRequest, response : HttpServletResponse) : void

```

Ilustración 114 SeeServlet.java

Tramita la petición de ver la solución de un caso de simulación

- **doGet():** Recibe la petición del usuario de ver la solución de un caso. Busca el HTML de este en su carpeta temporal y redirige al usuario a una nueva página para su visionado.

- **ShareServlet.java**

```

ShareServlet
#doGet(request : HttpServletRequest, response : HttpServletResponse) : void
#doPost(request : HttpServletRequest, response : HttpServletResponse) : void
#loadProjects(request : HttpServletRequest, response : HttpServletResponse) : void

```

Ilustración 115 ShareServlet

Servlet encargado de gestionar todo lo relacionado con las comparticiones de los casos de simulación.

- **doGet():** Recibe y gestiona las peticiones del usuario que quiere borrar una compartición. También se encarga de actualizar el panel de compartición de un caso tras un cambio llamado a *loadProjects()*.
- **doPost():** Se encarga de crear una nueva compartición en base al email pasado por el usuario y a la gestión del cambio de permisos en una compartición. Tras finalizar llama a *loadProjects()*.
- **loadProjects():** Realiza la actualización del estado de las comparticiones de un caso y redirige al panel de permisos de este.

- **UploadServlet.java**

```

UploadServlet
-TEMP_DIR : String
-SAVE_DIR : String
#doGet(request : HttpServletRequest, response : HttpServletResponse) : void
#doPost(request : HttpServletRequest, response : HttpServletResponse) : void
#insertFile(project : Project, user : User, path : String) : long

```

Ilustración 116 UploadServlet.java

Servlet que realizará la tarea de subida de los casos al servidor y lanzará su posterior simulación tras la subida.

- **doGet():** Su función es actuar en caso de que ocurra un problema de caso duplicado durante una subida. En base a la decisión del usuario mantiene las dos copias o borra la más antigua.
- **doPost():** Tramita la subida de un caso e inicia su posterior simulación. Llama a *insertFile()* para guardar los metadatos del caso en la base de datos.
- **insertFile():** Guarda la información del caso en la base de datos. Devuelve la ID asignada al proyecto.

7.6.1.3. Capa modelo

Las clases que engloban esta capa son los objetos DAO encargados de recoger la información de la base de datos y otras clases auxiliares con cierta lógica de negocio.

A continuación se ira clase por clase identificando y describiendo sus métodos y la finalidad general.

- **User.java**

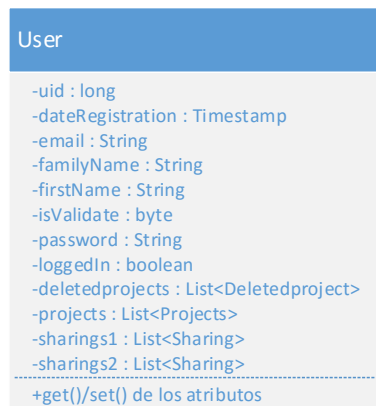


Ilustración 117 User.java

Esta clase es la encargada de recoger la información de un usuario cuando se consulta su información en la base de datos.

- **get()/set():** Métodos auxiliares para acceder o dar valor a los atributos del objeto.

- **Project.java**

Project
-idProject : long -dateCreation : Timestamp -dateModified : Timestamp -ONGFile : byte[] -projectDescription : String -projectName : String -statuscategory : Statuscategory -user : User -sharings : List<Sharing>
+get()/set() de los atributos

Ilustración 118 Project.java

Esta clase es la encargada de recoger la información de un caso cuando se consulta su información en la base de datos.

- **get()/set():** Métodos auxiliares para acceder o dar valor a los atributos del objeto.

- **Sharing.java**

Sharing
-idSharing : long -allowDelete : byte -allowDownload : byte -allowRecalculate : byte -allowShare : byte -dateChanged : Timestamp -dateShared : Timestamp -project : Project -user1 : User -user2 : User
+get()/set() de los atributos

Ilustración 119 Sharing.java

Esta clase es la encargada de recoger la información de una compartición cuando se consulta su información en la base de datos.

- **get()/set():** Métodos auxiliares para acceder o dar valor a los atributos del objeto.

- **Deletedproject.java**

Deletedproject
-idProject : long -dateCreation : Timestamp -dateDeleted : Timestamp -ONGFile : byte[] -projectDescription : String -projectName : String -statuscategory : Statuscategory -user : User
+get()/set() de los atributos

Ilustración 120 Deletedproject.java

Esta clase es la encargada de recoger la información de un caso borrado cuando se consulta su información en la base de datos.

- **get()/set():** Métodos auxiliares para acceder o dar valor a los atributos del objeto.

- **Statuscategory.java**

Statuscategory
-idProjectStatus : byte -statusDescription : String -statusName : String -deletedprojects : List<Deletedproject> -projects : List<Project>
+get()/set() de los atributos

Ilustración 121 Satatuscategory.java

Esta clase es la almacenada de recoger la información de un estado de un caso cuando se consulta su información en la base de datos.

- **get()/set():** Métodos auxiliares para acceder o dar valor a los atributos del objeto.

- **Proxymanager.java**

ProxyManager
-emf : EntityManagerFactory
+createObject(object : Object) : Object +deletedObject(object : Object): String +updateObject(object : Object) : Object +findObject (object Object) : Object +findObjects(object Object) : List<Object>

Ilustración 122 ProxyManager.java

Esta clase es la encargada de recoger todas las peticiones de acceso a la base de datos por parte del controlador y de devolver el resultado pedido.

- ***createObject()***: Recibe un objeto entidad, de los mencionados anteriormente, y lo inserta en la base de datos.
- ***deletedObject()***: Recibe un objeto entidad y elimina su equivalente de la base de datos.
- ***updateObject()***: Recibe un objeto entidad y actualiza su equivalente en la base de datos.
- ***findObject()*, *findObjects()***: Serie de funciones que servirán para devolver la información de las consultas necesarias a la base de datos para realizar una operación. Devuelven el resultado en un objeto entidad o en una lista de estos dependiendo de la consulta realizada.

- **Funciones.java**

```

Funciones

-SAVE_DIR : String
-TEMP_DIR : String
-----
+sendEmail(asunto : String, mensaje : String, to : String) : void
+cryptMD5(textoPlano : String) : String
+getFileName(path : String, p : Project) : void
+asignarPermiso(sharing : Sharing, perm : char) : void
+validarXML(path String, applicationPath : String) : boolean
+getManifest(ongf : Part, uid : long, applicationPath : String) : boolean
+extraerHTMLunico(applicationPath : String, uid : long, project : Project) : void
+extraerHTML(applicationPath : String, uid : long) : void
+validarResultadoHtml(name : String) : boolean
+modificarRutasContent(applicationPath : String, ruta : String, name : String) : void
+borrarDirectorio(directorio : File) : void
+borrarProject (project : Project, userBean : User) : void
+setLanguage(session : HttpSession) : void

```

Ilustración 123 Funciones.java

Esta clase auxiliar se encarga de realizar ciertas funciones necesarias para completar algunas tareas de la aplicación web.

- ***sendEmail()***: Recibe el asunto, contenido y destinatario del email y lo manda.
- ***cryptMD5()***: cifra con el algoritmo MD5 el texto recibido. Devuelve el resultado.
- ***getFileName()***: obtiene el nombre y la descripción del caso subido en base a las etiquetas de su manifest.xml. Guarda el resultado en el objeto Project recibido.
- ***asignarPermiso()***: recibe una compartición y el tipo de permiso cambiado. Actualiza la compartición en la base de datos.
- ***validarXML()***: valida que el manifest.xml del caso que se desea subir este correctamente estructurado.
- ***getManifest()***: busca el archivo manifest.xml en los directorios del caso a subir.
- ***extraerHTMLunico()***: extrae el HTML de un caso en particular.
- ***extraerHTML()***: extrae el HTML de todos los casos de la carpeta de un usuario.
- ***validarResultadoHtml()***: busca si el caso recién simulado tiene el archivo HTML con su solución.

- **modificarRutasContent()**: en caso de que la solución tenga rutas de imágenes o link a documentos o páginas modifica estas para adaptarlas y hacerlas accesibles desde la aplicación web.
- **borrarDirectorio()**: borra el directorio temporal de un usuario tras su cierre de sesión o invalidación de esta por timeout.
- **borrarProject()**: borra el archivo del servidor un caso borrado por el usuario desde la aplicación.
- **setLanguage()**: almacena en la sesión del usuario el lenguaje que hay activado actualmente.

- **Comunicacion.java**

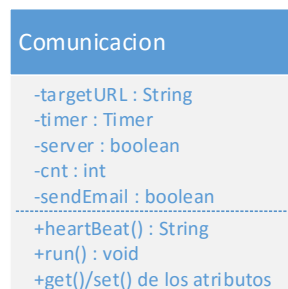
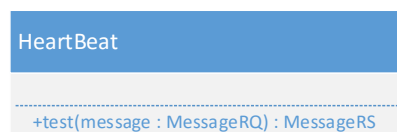


Ilustración 124 Comunicacion.java

Clase encargada de realizar el proceso de HeartBeat con el servidor de simulaciones.

- **heartBeat()**: función encargada de realizar el proceso de HeartBeat con el servidor de simulaciones. Devuelve la respuesta de este.
- **run()**: planificador del HeartBeat que se encarga de lanzar este proceso cada dos segundos llamando a **heartBeat()**. En caso de no recibir respuesta durante veinte segundos seguidos se encarga de enviar un email al administrador web.

- **HeartBeat.java**



Clase encargada de responder al servidor de simulaciones cuando este le realice su propio HeartBeat.

- **test()**: función encargada de realizar el proceso de HeartBeat con el servidor de simulaciones. Devuelve la respuesta a este.

- **Simulate.java**

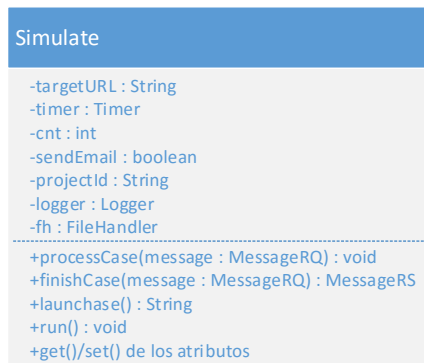


Ilustración 125 Simulate.java

Clase encargada de comunicarse con el servidor de simulaciones durante el proceso de simulación de un caso.

- **processCase():** recibe el mensaje del servidor de que el caso de simulación ha empezado a resolverse.
- **finishCase():** recibe el mensaje del servidor de que la simulación del caso concluyó. Devuelve una respuesta para hacerle saber que se dio por enterado.
- **launchCase():** envía al servidor la información del caso para planificar la simulación de este.
- **run():** planificador de la simulación que llamará a *launchCase()* cada cinco segundos hasta recibir la respuesta del servidor de que el caso está en la cola de simulaciones. En caso de no recibir respuesta durante cincuenta segundos manda un email al administrador web.

- **MessageRQ.java**



Ilustración 126 MessageRQ.java

Clase objeto encargada de almacenar la información de los mensajes enviados al servidor de simulaciones.

- **MessageRS.java**

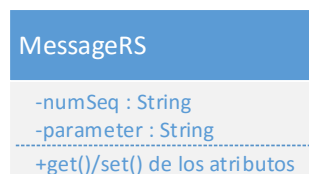


Ilustración 127 MessageRS.java

Clase objeto encargada de almacenar la información de los mensajes respuesta del servidor de simulaciones.

7.6.2. Diseño de asociaciones y agregaciones

En esta tarea se completan las asociaciones entre las clases del modelo de clases de diseño en base a lo recogido previamente en el análisis y en estos apartados anteriores.

Para ello se remite al lector a los puntos 6.7. *Análisis de clases* y 6.7.2. *Identificación de asociaciones y agregaciones* del análisis. Se seguirá la misma estructura salvo cambiando el nombre de las clases por el de las propuestas en el diseño en los casos que sea necesario.

7.6.3. Identificación de atributos de las clases

El objetivo de esta tarea es identificar y describir los atributos de las clases de diseño. Para ello se expondrá a continuación una tabla por cada clase con atributos identificada en los apartados anteriores.

SeeServlet.java		
TEMP_DIR	Descripción	Directorio dentro del servidor donde son almacenados los archivos temporales de los usuarios
	Tipo	String

Tabla 172 atributos SeeServlet.java

UploadServlet.java		
TEMP_DIR	Descripción	Directorio dentro del servidor donde es almacenado el HTML de los casos de los usuarios
	Tipo	String
SAVE_DIR	Descripción	Directorio dentro del servidor donde son almacenados los casos
	Tipo	String

Tabla 173 atributos UploadServlet.java

User.java		
uid	Descripción	ID única por la que se identifica a un usuario
	Tipo	Long
dateRegistration	Descripción	Fecha de registro del usuario en la aplicación
	Tipo	Timestamp
email	Descripción	Email de registro del usuario
	Tipo	String
familyName	Descripción	Apellido/s del usuario
	Tipo	String
firstName	Descripción	Nombre del usuario
	Tipo	String
isValidate	Descripción	Identificador de si la cuenta está activada
	Tipo	byte
password	Descripción	Contraseña del usuario (cifrada en MD5)
	Tipo	String

loggedIn	Descripción	Marca que indica si el usuario esta logueado en ese momento o no
	Tipo	boolean
deletedProjects	Descripción	Lista de casos borrados ligados a ese usuario
	Tipo	List<Deletedproject>
projects	Descripción	Lista de casos ligados a ese usuario
	Tipo	List<Project>
sharings1	Descripción	Lista de comparticiones que le han dado a ese usuario
	Tipo	List<Sharing>
sharings2	Descripción	Lista de comparticiones de las que es dueño ese usuario
	Tipo	List<Sharing>

Tabla 174 atributos User.java

Project.java		
idProject	Descripción	ID única por la que se identifica a un caso
	Tipo	Long
dateCreation	Descripción	Fecha en la que el caso fue subido al servidor
	Tipo	Timestamp
dateModified	Descripción	Fecha de última simulación del caso
	Tipo	Timestamp
ONGFile	Descripción	Path del caso en el servidor
	Tipo	byte[]
projectDescription	Descripción	Descripción del caso, incluida en el manifest.xml
	Tipo	String
projectName	Descripción	Nombre del caso, incluida en el manifest.xml
	Tipo	String
statuscategory	Descripción	Estado del caso
	Tipo	Statuscategory
user	Descripción	Usuario propietario del caso
	Tipo	User
sharings	Descripción	Lista de comparticiones que existen sobre este caso
	Tipo	List<Sharing>

Tabla 175 atributos Project.java

Sharing.java		
idSharing	Descripción	ID única por la que se identifica a una compartición
	Tipo	Long
allowDelete	Descripción	Permiso de borrar el caso
	Tipo	byte
allowDownload	Descripción	Permiso de descargar el caso
	Tipo	byte
allowRecalculate	Descripción	Permiso de resimular el caso
	Tipo	byte
allowShare	Descripción	Permiso de compartir el caso
	Tipo	byte
dateChanged	Descripción	Fecha de última modificación de los permisos
	Tipo	Timestamp

dateShared	Descripción	Fecha de creación de la compartición
	Tipo	Timestamp
project	Descripción	Caso ligado a esa compartición
	Tipo	Project
user1	Descripción	Usuario al que se le ha compartido el caso
	Tipo	User
user2	Descripción	Usuario que compartió el caso, no necesariamente el dueño de él
	Tipo	User

Tabla 176 atributos Sharing.java

Deletedproject.java		
idProject	Descripción	ID única por la que se identifica a un caso
	Tipo	Long
dateCreation	Descripción	Fecha en la que el caso fue subido al servidor
	Tipo	Timestamp
dateDeleted	Descripción	Fecha de borrado del caso
	Tipo	Timestamp
ONGFile	Descripción	Path del caso en el servidor
	Tipo	byte[]
projectDescription	Descripción	Descripción del caso, incluida en el manifest.xml
	Tipo	String
projectName	Descripción	Nombre del caso, incluida en el manifest.xml
	Tipo	String
statuscategory	Descripción	Estado del caso
	Tipo	Statuscategory
user	Descripción	Usuario propietario del caso
	Tipo	User

Tabla 177 atributos Deletedproject.java

Statuscategory.java		
idProjectStatus	Descripción	Valor del estado (0- pendiente, 1-calculando, 2-simulado, 3-errores)
	Tipo	byte
statusDescription	Descripción	Descripción del estado del caso
	Tipo	String
statusName	Descripción	Nombre del estado
	Tipo	String
deletedProjects	Descripción	Lista de casos borrados con estado
	Tipo	List<deletedProject>
projects	Descripción	Lista de casos con estado
	Tipo	List<Project>

Tabla 178 atributos Statuscategory.java

ProxyManager.java		
emf	Descripción	Gestor de las entidades JPA que usaremos para interactuar con la base de datos
	Tipo	EntityManagerFactory

Tabla 179 atributos ProxyManager.java

Funciones.java		
----------------	--	--

TEMP_DIR	Descripción	Directorio dentro del servidor donde es almacenado el HTML de los casos de los usuarios
	Tipo	String
SAVE_DIR	Descripción	Directorio dentro del servidor donde son almacenados los casos
	Tipo	String

Tabla 180 atributos Funciones.java

Comunicacion.java		
targerURL	Descripción	URL del destinatario del WebServices
	Tipo	String
timer	Descripción	Planificador que ejecutará el servicio cada dos segundos
	Tipo	Timer
server	Descripción	Marca que indicará el estado en el que se encuentra el servidor de simulaciones
	Tipo	boolean
cnt	Descripción	Contador para gestionar la cantidad de veces que fallo la comunicación con el servidor
	Tipo	int
sendEmail	Descripción	Marca que nos indicará si se ha enviado un email o no al administrador en caso de fallo en las comunicaciones
	Tipo	boolean

Tabla 181 atributos Comunicacion.java

Simulate.java		
targerURL	Descripción	URL del destinatario del WebServices
	Tipo	String
timer	Descripción	Planificador que ejecutará el servicio cada cinco segundos
	Tipo	Timer
cnt	Descripción	Contador para gestionar la cantidad de veces que fallo la comunicación con el servidor
	Tipo	int
sendEmail	Descripción	Marca que nos indicará si se ha enviado un email o no al administrador en caso de fallo en las comunicaciones
	Tipo	Boolean
projectId	Descripción	ID del caso a simular
	Tipo	String
logger	Descripción	Log que recogerá las etapas de la simulación del caso
	Tipo	Logger
fh	Descripción	Archivo que guardará el log generado durante la simulación
	Tipo	FileHandler

Tabla 182 atributos Simulate.java

MessageRQ.java		
numSeq	Descripción	Numero identificativo del mensaje. Deberán ser iguales el del mensaje solicitud y respuesta

	Tipo	String
	Descripción	Instrucción dada al destinatario y que se quiere realizar
command	Tipo	String
	Descripción	Valores necesarios para llevar a cabo la tarea
parameter	Tipo	String

Tabla 183 atributos MessageRQ.java

MessageRS.java		
numSeq	Descripción	Numero identificativo del mensaje. Deberán ser iguales el del mensaje solicitud y respuesta
	Tipo	String
parameter	Descripción	Valores necesarios para llevar a cabo la tarea
	Tipo	String

Tabla 184 atributos MessageRS.java

7.6.4. Identificación de los métodos de las clases

El objetivo es definir detalladamente las operaciones que puede realizar cada clase de diseño. Para ello se toma como punto de partida el modelo de clases generado hasta el momento, así como el diseño de los casos de uso reales.

En el apartado 7.6.1. *Identificación de clases de diseño* de este diseño de clases se ha hecho una breve introducción de las funcionalidades de cada una de las clases de la aplicación web. A continuación se hará una descripción más detallada, se adjuntará una tabla por clase en la que se explicarán sus métodos.

ServletController.java			
doPost			
Tipo	Método	Visibilidad	<input type="checkbox"/> Pública <input type="checkbox"/> Privada <input checked="" type="checkbox"/> Protegida
Descripción	Recibe las solicitudes que llegan por el método POST de Http		
Parámetros	De Entrada:		
	<ul style="list-style-type: none">• request: Petición de Http• response: Respuesta de Http		
	De Salida:		
	<ul style="list-style-type: none">• Ninguno.		
Excepciones	ServletException, IOException		
doGet			
Tipo	Método	Visibilidad	<input type="checkbox"/> Pública <input type="checkbox"/> Privada <input checked="" type="checkbox"/> Protegida
Descripción	Recibe las solicitudes que llegan por el método GET de Http		
Parámetros	De Entrada:		
	<ul style="list-style-type: none">• request: Petición de Http• response: Respuesta de Http		
	De Salida:		
	<ul style="list-style-type: none">• Ninguno.		
Excepciones	ServletException, IOException		

procesRequest			
Tipo	Método	Visibilidad	<input type="checkbox"/> Pública <input type="checkbox"/> Privada <input checked="" type="checkbox"/> Protegida
Descripción	Comprueba que la petición es buena y la redirige al servlet correcto		
Parámetros	De Entrada: <ul style="list-style-type: none">• request: Petición de Http• response: Respuesta de Http De Salida: <ul style="list-style-type: none">• Redirige al servlet correspondiente que debe tratar la petición.		
Excepciones	ServletException, IOException		
init			
Tipo	Método	Visibilidad	<input type="checkbox"/> Pública <input type="checkbox"/> Privada <input checked="" type="checkbox"/> Protegida
Descripción	Inicia el proceso de heartbeat con el servidor de simulaciones		
Parámetros	De Entrada: <ul style="list-style-type: none">• config: Configuración del servlet De Salida: <ul style="list-style-type: none">• Ninguno.		
Excepciones	ServletException		

Tabla 185 métodos ServletController.java

AccountServlet.java			
doPost			
Tipo	Método	Visibilidad	<input type="checkbox"/> Pública <input type="checkbox"/> Privada <input checked="" type="checkbox"/> Protegida
Descripción	Actualiza los datos de registro de un usuario		
Parámetros	De Entrada: <ul style="list-style-type: none"> request: Petición de Http response: Respuesta de Http De Salida: <ul style="list-style-type: none"> Redirige al panel de cuenta del usuario actualizado. 		
Excepciones	ServletException, IOException		

Tabla 186 métodos AccountServlet.java

ActivateServlet.java			
doGet			
Tipo	Método	Visibilidad	<input type="checkbox"/> Pública <input type="checkbox"/> Privada <input checked="" type="checkbox"/> Protegida
Descripción	Tramita la activación de la cuenta de un usuario recién registrado		
Parámetros	De Entrada: <ul style="list-style-type: none"> request: Petición de Http response: Respuesta de Http De Salida: <ul style="list-style-type: none"> Ninguno. 		
Excepciones	ServletException, IOException		

Tabla 187 métodos ActivateServlet.java

DeleteServlet.java			
doGet			

Tipo	Método	Visibilidad	<input type="checkbox"/> Pública <input type="checkbox"/> Privada <input checked="" type="checkbox"/> Protegida
Descripción	Gestiona el borrado de un caso de simulación		
Parámetros	De Entrada: <ul style="list-style-type: none"> • request: Petición de Http • response: Respuesta de Http De Salida: <ul style="list-style-type: none"> • Redirige a la página principal del repositorio. 		
Excepciones	ServletException, IOException		

Tabla 188 métodos DeletedServlet.java

DownloadServlet.java			
doGet			
Tipo	Método	Visibilidad	<input type="checkbox"/> Pública <input type="checkbox"/> Privada <input checked="" type="checkbox"/> Protegida
Descripción	Prepara la descarga de un caso de simulación		
Parámetros	De Entrada: <ul style="list-style-type: none">• request: Petición de Http• response: Respuesta de Http De Salida: <ul style="list-style-type: none">• Ninguno.		
Excepciones	ServletException, IOException		
descargar			
Tipo	Método	Visibilidad	<input type="checkbox"/> Pública <input type="checkbox"/> Privada <input checked="" type="checkbox"/> Protegida
Descripción	Realiza la descarga de un caso de simulación		
Parámetros	De Entrada: <ul style="list-style-type: none">• request: Petición de Http• response: Respuesta de Http• project: caso de simulación De Salida: <ul style="list-style-type: none">• Inicia la descarga del archivo del caso.		
Excepciones	ServletException, IOException		

Tabla 189 métodos DownloadServlet.java

LanguageServlet.java			
doGet			
Tipo	Método	Visibilidad	<input type="checkbox"/> Pública <input type="checkbox"/> Privada <input checked="" type="checkbox"/> Protegida
Descripción	Realiza el cambio de idioma en la aplicación web		
Parámetros	De Entrada: <ul style="list-style-type: none"> • request: Petición de Http • response: Respuesta de Http De Salida: <ul style="list-style-type: none"> • Redirige a la página actual con el idioma cambiado. 		
Excepciones	ServletException, IOException		

Tabla 190 métodos LanguageServlet.java

LoginServlet.java

doGet			
Tipo	Método	Visibilidad	<input type="checkbox"/> Pública <input type="checkbox"/> Privada <input checked="" type="checkbox"/> Protegida
Descripción	Realiza la actualización del repositorio y el control de las páginas de casos que se tienen de un usuario		
Parámetros	De Entrada: <ul style="list-style-type: none">• request: Petición de Http• response: Respuesta de Http De Salida: <ul style="list-style-type: none">• Redirige a la página principal del repositorio.		
Excepciones	ServletException, IOException		
doPost			
Tipo	Método	Visibilidad	<input type="checkbox"/> Pública <input type="checkbox"/> Privada <input checked="" type="checkbox"/> Protegida
Descripción	Realiza la actualización del repositorio, el control de las páginas de casos que se tienen y la descompresión del HTML de los casos de un usuario		
Parámetros	De Entrada: <ul style="list-style-type: none">• request: Petición de Http• response: Respuesta de Http De Salida: <ul style="list-style-type: none">• Redirige a la página principal del repositorio.		
Excepciones	ServletException, IOException		

Tabla 191 métodos LoginServlet.java

LogoutServlet.java			
doGet			
Tipo	Método	Visibilidad	<input type="checkbox"/> Pública <input type="checkbox"/> Privada <input checked="" type="checkbox"/> Protegida
Descripción	Realiza el cierre de sesión del usuario y el borrado de la carpeta temporal de este con el HTML de sus casos		
Parámetros	De Entrada: <ul style="list-style-type: none"> request: Petición de Http response: Respuesta de Http De Salida: <ul style="list-style-type: none"> Redirige a la página inicial de la aplicación web. 		
Excepciones	ServletException, IOException, NullPointerException		

Tabla 192 métodos LogoutServlet.java

RunServlet.java			
doGet			
Tipo	Método	Visibilidad	<input type="checkbox"/> Pública <input type="checkbox"/> Privada <input checked="" type="checkbox"/> Protegida
Descripción	Da inicio a la simulación de un caso		
Parámetros	De Entrada: <ul style="list-style-type: none"> request: Petición de Http response: Respuesta de Http De Salida: <ul style="list-style-type: none"> Redirige a la página del repositorio. 		
Excepciones	ServletException, IOException		

Tabla 193 métodos RunServlet.java

RegisterServlet.java		
doPost		
Tipo	Método	Visibilidad <input type="checkbox"/> Pública <input type="checkbox"/> Privada <input checked="" type="checkbox"/> Protegida
Descripción	Tramita el registro de un nuevo usuario y manda el email de activación de la cuenta	
Parámetros	De Entrada: <ul style="list-style-type: none"> request: Petición de Http response: Respuesta de Http De Salida: <ul style="list-style-type: none"> Manda email de activación. 	
Excepciones	ServletException, IOException	

Tabla 194 métodos RegisterServlet.java

PageServlet.java		
doGet		
Tipo	Método	Visibilidad <input type="checkbox"/> Pública <input type="checkbox"/> Privada <input checked="" type="checkbox"/> Protegida
Descripción	Realiza la gestión y carga de los casos de simulación que hay en cada página del repositorio de un usuario	
Parámetros	De Entrada: <ul style="list-style-type: none"> request: Petición de Http response: Respuesta de Http De Salida: <ul style="list-style-type: none"> Redirige a la página de casos del repositorio solicitada por el usuario. 	
Excepciones	ServletException, IOException	

Tabla 195 métodos PageServlet.java

RecoverServlet.java		
doGet		
Tipo	Método	Visibilidad <input type="checkbox"/> Pública <input type="checkbox"/> Privada <input checked="" type="checkbox"/> Protegida
Descripción	Comprueba que el link de recuperación usado por el usuario es correcto	
Parámetros	De Entrada: <ul style="list-style-type: none">request: Petición de Httpresponse: Respuesta de Http De Salida: <ul style="list-style-type: none">Redirige a la página donde el usuario elegirá su nueva contraseña.	
Excepciones	ServletException, IOException	
doPost		
Tipo	Método	Visibilidad <input type="checkbox"/> Pública <input type="checkbox"/> Privada <input checked="" type="checkbox"/> Protegida
Descripción	Manda el email de recuperación de contraseña y gestiona la modificación de esta	
Parámetros	De Entrada: <ul style="list-style-type: none">request: Petición de Http	

	<ul style="list-style-type: none"> • response: Respuesta de Http De Salida: <ul style="list-style-type: none"> • Manda email de recuperación o redirige al index de la web según la operación realizada.
Excepciones	ServletException, IOException

Tabla 196 métodos RecoverServlet.java

SeeServlet.java	
doGet	
Tipo	Método Visibilidad <input type="checkbox"/> Pública <input type="checkbox"/> Privada <input checked="" type="checkbox"/> Protegida
Descripción	Prepara el visionado de la solución de un caso simulado
Parámetros	De Entrada: <ul style="list-style-type: none"> • request: Petición de Http • response: Respuesta de Http De Salida: <ul style="list-style-type: none"> • Redirige a la página con la solución del caso.
Excepciones	ServletException, IOException

Tabla 197 métodos SeeServlet.java

ShareServlet.java	
doGet	
Tipo	Método Visibilidad <input type="checkbox"/> Pública <input type="checkbox"/> Privada <input checked="" type="checkbox"/> Protegida
Descripción	Gestiona el borrado de comparticiones y las cargas del panel de comparticiones de un caso
Parámetros	De Entrada: <ul style="list-style-type: none"> • request: Petición de Http • response: Respuesta de Http De Salida: <ul style="list-style-type: none"> • Ninguno.
Excepciones	ServletException, IOException
doPost	
Tipo	Método Visibilidad <input type="checkbox"/> Pública <input type="checkbox"/> Privada <input checked="" type="checkbox"/> Protegida
Descripción	Gestiona la creación de nuevas comparticiones y el cambio de permisos
Parámetros	De Entrada: <ul style="list-style-type: none"> • request: Petición de Http • response: Respuesta de Http De Salida: <ul style="list-style-type: none"> • Ninguno.
Excepciones	ServletException, IOException
loadProjects	
Tipo	Método Visibilidad <input type="checkbox"/> Pública <input type="checkbox"/> Privada <input checked="" type="checkbox"/> Protegida
Descripción	Prepara la carga de comparticiones de un caso para el panel de compartición
Parámetros	De Entrada: <ul style="list-style-type: none"> • request: Petición de Http

	<ul style="list-style-type: none"> • response: Respuesta de Http De Salida: <ul style="list-style-type: none"> • Redirige al panel de compartición actualizado de ese caso.
Excepciones	ServletException, IOException

Tabla 198 métodos ShareServlet.java

UploadServlet.java			
doGet			
Tipo	Método	Visibilidad	<input type="checkbox"/> Pública <input type="checkbox"/> Privada <input checked="" type="checkbox"/> Protegida
Descripción	Realiza la comprobación de duplicados a la hora de subir nuevos casos		
Parámetros	De Entrada: <ul style="list-style-type: none">• request: Petición de Http• response: Respuesta de Http De Salida: <ul style="list-style-type: none">• Ninguno.		
Excepciones	ServletException, IOException		
doPost			
Tipo	Método	Visibilidad	<input type="checkbox"/> Pública <input type="checkbox"/> Privada <input checked="" type="checkbox"/> Protegida
Descripción	Realiza la subida de un caso nuevo al servidor		
Parámetros	De Entrada: <ul style="list-style-type: none">• request: Petición de Http• response: Respuesta de Http De Salida: <ul style="list-style-type: none">• Ninguno.		
Excepciones	ServletException, IOException		
insertFile			
Tipo	Método	Visibilidad	<input type="checkbox"/> Pública <input type="checkbox"/> Privada <input checked="" type="checkbox"/> Protegida
Descripción	Escribe la información del nuevo caso en la base de datos		
Parámetros	De Entrada: <ul style="list-style-type: none">• project: caso a subir• user: usuario que lo está subiendo• path: localización del caso en el servidor De Salida: <ul style="list-style-type: none">• ID del caso.		
Excepciones	Ninguna		

Tabla 199 UploadServlet.java

User.java			
getAtributo			
Tipo	Método	Visibilidad	<input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Devuelve el valor del atributo de la entidad consultado		
Parámetros	De Entrada: <ul style="list-style-type: none"> • Ninguno De Salida: <ul style="list-style-type: none"> • Atributo consultado. 		

Excepciones	Ninguna		
setAtributo			
Tipo	Método	Visibilidad	<input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Actualiza el valor del atributo de la entidad deseado		
Parámetros	De Entrada: <ul style="list-style-type: none">• atributo: valor nuevo del atributo		
	De Salida: <ul style="list-style-type: none">• Ninguno.		
Excepciones	Ninguna		

Tabla 200 métodos User.java

Project.java			
getAtributo			
Tipo	Método	Visibilidad	<input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Devuelve el valor del atributo de la entidad consultado		
Parámetros	De Entrada: <ul style="list-style-type: none">Ninguno De Salida: <ul style="list-style-type: none">Atributo consultado.		
Excepciones	Ninguna		
setAtributo			
Tipo	Método	Visibilidad	<input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Actualiza el valor del atributo de la entidad deseado		
Parámetros	De Entrada: <ul style="list-style-type: none">atributo: valor nuevo del atributo De Salida: <ul style="list-style-type: none">Ninguno.		
Excepciones	Ninguna		

Tabla 201 métodos Project.java

Sharing.java			
getAtributo			
Tipo	Método	Visibilidad	<input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Devuelve el valor del atributo de la entidad consultado		
Parámetros	De Entrada: <ul style="list-style-type: none">• Ninguno De Salida: <ul style="list-style-type: none">• Atributo consultado.		
Excepciones	Ninguna		
setAtributo			
Tipo	Método	Visibilidad	<input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Actualiza el valor del atributo de la entidad deseado		
Parámetros	De Entrada: <ul style="list-style-type: none">• atributo: valor nuevo del atributo De Salida:		

	<ul style="list-style-type: none"> Ninguno.
Excepciones	Ninguna

Tabla 202 métodos Sharing.java

Deletedproject.java			
getAtributo			
Tipo	Método	Visibilidad	<input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Devuelve el valor del atributo de la entidad consultado		
Parámetros	De Entrada: <ul style="list-style-type: none">Ninguno De Salida: <ul style="list-style-type: none">Atributo consultado.		
Excepciones	Ninguna		
setAtributo			
Tipo	Método	Visibilidad	<input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Actualiza el valor del atributo de la entidad deseado		
Parámetros	De Entrada: <ul style="list-style-type: none">atributo: valor nuevo del atributo De Salida: <ul style="list-style-type: none">Ninguno.		
Excepciones	Ninguna		

Tabla 203 métodos Deletedproject.java

Statuscategory.java			
getAtributo			
Tipo	Método	Visibilidad	<input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Devuelve el valor del atributo de la entidad consultado		
Parámetros	De Entrada: <ul style="list-style-type: none">Ninguno De Salida: <ul style="list-style-type: none">Atributo consultado.		
Excepciones	Ninguna		
setAtributo			
Tipo	Método	Visibilidad	<input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Actualiza el valor del atributo de la entidad deseado		
Parámetros	De Entrada: <ul style="list-style-type: none">atributo: valor nuevo del atributo De Salida: <ul style="list-style-type: none">Ninguno.		
Excepciones	Ninguna		

Tabla 204 métodos Statuscategory.java

ProxyManager.java			
createObject			
Tipo	Método	Visibilidad	<input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida

Descripción	Inserta en la base de datos el objeto pasado		
Parámetros	De Entrada: <ul style="list-style-type: none">• objetc: objeto a insertar		
	De Salida: <ul style="list-style-type: none">• Devuelve el mismo objeto con todos los valores.		
Excepciones	Exception		
deletedObject			
Tipo	Método	Visibilidad	<input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Borra el registro de la base de datos en base al objeto pasado		
Parámetros	De Entrada: <ul style="list-style-type: none">• objetc: objeto a borrar		
	De Salida: <ul style="list-style-type: none">• Devuelve un string con la confirmación de que fue bien		
Excepciones	Exception		
updateObject			
Tipo	Método	Visibilidad	<input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Actualiza el registro en la base de datos en base al objeto pasado		
Parámetros	De Entrada: <ul style="list-style-type: none">• objetc: objeto a actualizar		
	De Salida: <ul style="list-style-type: none">• Devuelve el mismo objeto		
Excepciones	Exception		
findObject			
Tipo	Método	Visibilidad	<input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Busca un registro en la base de datos en base a los valores pasados		
Parámetros	De Entrada: <ul style="list-style-type: none">• Info: valores de las columnas del registro a buscar		
	De Salida: <ul style="list-style-type: none">• Objeto que coincide con los valores buscados.		
Excepciones	Ninguna		
findObjects			
Tipo	Método	Visibilidad	<input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Busca unos registros en la base de datos en base a los valores pasados		
Parámetros	De Entrada: <ul style="list-style-type: none">• Info: valores de las columnas de los registros a buscar		
	De Salida: <ul style="list-style-type: none">• Lista de objetos que coinciden con los valores buscados.		
Excepciones	Ninguna		

Tabla 205 métodos ProxyManager.java

Funciones.java			
sendEmail			
Tipo	Método	Visibilidad	<input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Manda un email		

Parámetros	De Entrada:		
	<ul style="list-style-type: none">• asunto: asunto del correo• mensaje: contenido del mensaje• to: email del destinatario		
	De Salida:		
	<ul style="list-style-type: none">• Manda el email.		
Excepciones	MessagingException		
cryptMD5			
Tipo	Método	Visibilidad	<input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Cifra un texto mediante el algoritmo MD5		
Parámetros	De Entrada:		
	<ul style="list-style-type: none">• textoPlano: texto a cifrar		
	De Salida:		
	<ul style="list-style-type: none">• Texto cifrado.		
Excepciones	NoSuchAlgorithmException		
getFileName			
Tipo	Método	Visibilidad	<input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Obtiene la descripción y el nombre de un caso a partir de su manifest.xml		
Parámetros	De Entrada:		
	<ul style="list-style-type: none">• path: ruta en el servidor del caso• p: objeto caso		
	De Salida:		
	<ul style="list-style-type: none">• Se guardan en el objeto caso los valores del nombre y descripción.		
Excepciones	XPathExpressionException, ParserConfigurationException, SAXException, IOException		
asignarPermiso			
Tipo	Método	Visibilidad	<input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Actualiza los permisos de una compartición		
Parámetros	De Entrada:		
	<ul style="list-style-type: none">• sharing: compartición a actualizar• perm: permiso modificado		
	De Salida:		
	<ul style="list-style-type: none">• Se actualiza el permiso de la compartición pasada.		
Excepciones	Ninguna		
validarXML			
Tipo	Método	Visibilidad	<input type="checkbox"/> Pública <input checked="" type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Valida el contenido del manifest.xml		
Parámetros	De Entrada:		
	<ul style="list-style-type: none">• path: ruta en el servidor del caso• aplicationPath: ruta absoluta de la aplicación web		
	De Salida:		
	<ul style="list-style-type: none">• Booleano que confirmará si es correcto o no.		
Excepciones	SAXException, ParserConfigurationException, IOException		

getManifest			
Tipo	Método	Visibilidad	<input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Comprueba si un caso tiene el archivo manifest.xml		
Parámetros	De Entrada: <ul style="list-style-type: none">• ongf: archivo a subir• uid: ID del usuario propietario• applicationPath: ruta absoluta de la aplicación web De Salida: <ul style="list-style-type: none">• Booleano que confirmará si existe o no.		
Excepciones	IOException		
extraerHTMLUnico			
Tipo	Método	Visibilidad	<input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Extrae el HTML de un único caso de simulación		
Parámetros	De Entrada: <ul style="list-style-type: none">• applicationPath: ruta absoluta de la aplicación web• uid: ID del usuario propietario• project: objeto caso de simulación De Salida: <ul style="list-style-type: none">• El HTML es extraído en la carpeta temporal del usuario.		
Excepciones	FileNotFoundException, IOException		
extraerHTML			
Tipo	Método	Visibilidad	<input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Extrae el HTML de todos los casos de la carpeta de un usuario		
Parámetros	De Entrada: <ul style="list-style-type: none">• applicationPath: ruta absoluta de la aplicación web• uid: ID del usuario propietario De Salida: <ul style="list-style-type: none">• El HTML es extraído en la carpeta temporal del usuario.		
Excepciones	FileNotFoundException, IOException		
validarResultadoHtml			
Tipo	Método	Visibilidad	<input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Comprueba si un caso ya simulado contiene el HTML con su solución		
Parámetros	De Entrada: <ul style="list-style-type: none">• name: nombre del HTML donde se supone que está la solución De Salida: <ul style="list-style-type: none">• Booleano que confirmará si es correcto o no.		
Excepciones	Ninguna		
modificarRutasContent			
Tipo	Método	Visibilidad	<input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Si el HTML de la solución de un caso contiene imágenes o rutas a archivos internos este método se encargará de modificarlas para hacerlas compatibles con la aplicación		
Parámetros	De Entrada: <ul style="list-style-type: none">• applicationPath: ruta absoluta de la aplicación web		

	<ul style="list-style-type: none"> • ruta: path del archivo del caso descomprimido en el servidor • name: nombre del archivo HTML a modificar De Salida: <ul style="list-style-type: none"> • El archivo HTML es modificado si hay necesidad.
Excepciones	FileNotFoundException, UnsupportedEncodingException, IOException
borrarDirectorio	
Tipo	Método Visibilidad <input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Borra el contenido y directorio temporal de un usuario
Parámetros	De Entrada: <ul style="list-style-type: none"> • directorio: directorio a borrar De Salida: <ul style="list-style-type: none"> • El directorio es borrado.
Excepciones	Ninguna
borrarProject	
Tipo	Método Visibilidad <input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Borrar el archivo en el servidor de un caso
Parámetros	De Entrada: <ul style="list-style-type: none"> • project: objeto caso con la información para el borrado • userBean: propietario del caso De Salida: <ul style="list-style-type: none"> • El caso es borrado.
Excepciones	FileNotFoundException, IOException
setLanguage	
Tipo	Método Visibilidad <input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Actualiza el lenguaje en la sesión del usuario
Parámetros	De Entrada: <ul style="list-style-type: none"> • sesion: sesión del usuario De Salida: <ul style="list-style-type: none"> • El idioma es cambiado.
Excepciones	Ninguna

Tabla 206 métodos Funciones.java

Comunicacion.java	
heartBeat	
Tipo	Método Visibilidad <input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Lanza el heartbeat contra el servidor de simulaciones
Parámetros	De Entrada: <ul style="list-style-type: none"> • Ninguno De Salida: <ul style="list-style-type: none"> • Mensaje respuesta del servidor de simulaciones.
Excepciones	RuntimeException, MalformedURLException, IOException
run	
Tipo	Método Visibilidad <input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Planificador del heartbeat

Parámetros	De Entrada: <ul style="list-style-type: none">Ninguno De Salida: <ul style="list-style-type: none">Manda email al administrador web si el heartbeat falla diez veces seguidas.		
Excepciones	Ninguna		
getAtributo			
Tipo	Método	Visibilidad	<input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Devuelve el valor del atributo de la entidad consultado		
Parámetros	De Entrada: <ul style="list-style-type: none">Ninguno De Salida: <ul style="list-style-type: none">Atributo consultado.		
Excepciones	Ninguna		
setAtributo			
Tipo	Método	Visibilidad	<input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Actualiza el valor del atributo de la entidad deseado		
Parámetros	De Entrada: <ul style="list-style-type: none">atributo: valor nuevo del atributo De Salida: <ul style="list-style-type: none">Ninguno.		
Excepciones	Ninguna		

Tabla 207 métodos Comunicacion.java

HeartBeat.java			
test			
Tipo	Método	Visibilidad	<input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Recibe la solicitud de heartbeat del servidor de simulaciones		
Parámetros	De Entrada:		
	<ul style="list-style-type: none"> message: MessageRQ enviado por el servidor de simulaciones 		
Excepciones	De Salida:		
	<ul style="list-style-type: none"> MessageRS de respuesta al servidor de simulaciones. 		

Tabla 208 métodos HeartBeat.java

Simulate.java			
procesCase			
Tipo	Método	Visibilidad	<input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Recibe el mensaje del servidor de simulaciones de que la simulación del caso fue lanzada		
Parámetros	De Entrada:		
	<ul style="list-style-type: none"> message: MessageRQ enviado por el servidor de simulaciones 		
Excepciones	De Salida:		
	<ul style="list-style-type: none"> Ninguno. 		

finishCase			
Tipo	Método	Visibilidad	<input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Recibe el mensaje del servidor de simulaciones de que la simulación del caso ha finalizado		
Parámetros	De Entrada: <ul style="list-style-type: none">• message: MessageRQ enviado por el servidor de simulaciones De Salida: <ul style="list-style-type: none">• MessageRS de respuesta al servidor de simulaciones.		
Excepciones	Ninguna		
launchase			
Tipo	Método	Visibilidad	<input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Envía la petición de simulación de un caso al servidor de simulaciones		
Parámetros	De Entrada: <ul style="list-style-type: none">• Ninguno De Salida: <ul style="list-style-type: none">• Respuesta del servidor.		
Excepciones	RuntimeException, MalformedURLException, IOException		
run			
Tipo	Método	Visibilidad	<input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Planificador de la simulación del caso		
Parámetros	De Entrada: <ul style="list-style-type: none">• Ninguno De Salida: <ul style="list-style-type: none">• Email al administrador web si hay problemas para notificar la petición de simulación al servidor de simulaciones.		
Excepciones	Ninguna		
getAtributo			
Tipo	Método	Visibilidad	<input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Devuelve el valor del atributo de la entidad consultado		
Parámetros	De Entrada: <ul style="list-style-type: none">• Ninguno De Salida: <ul style="list-style-type: none">• Atributo consultado..		
Excepciones	Ninguna		
setAtributo			
Tipo	Método	Visibilidad	<input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Actualiza el valor del atributo de la entidad deseado		
Parámetros	De Entrada: <ul style="list-style-type: none">• atributo: valor nuevo del atributo De Salida: <ul style="list-style-type: none">• Ninguno.		
Excepciones	Ninguna		

Tabla 209 Simulate.java

MessageRQ.java			
getAtributo			
Tipo	Método	Visibilidad	<input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Devuelve el valor del atributo de la entidad consultado		
Parámetros	De Entrada: <ul style="list-style-type: none">Ninguno De Salida: <ul style="list-style-type: none">Atributo consultado.		
Excepciones	Ninguna		
setAttribute			
Tipo	Método	Visibilidad	<input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Actualiza el valor del atributo de la entidad deseado		
Parámetros	De Entrada: <ul style="list-style-type: none">atributo: valor nuevo del atributo De Salida: <ul style="list-style-type: none">Ninguno.		
Excepciones	Ninguna		

Tabla 210 métodos MessageRQ.java

MessageRS.java			
getAtributo			
Tipo	Método	Visibilidad	<input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Devuelve el valor del atributo de la entidad consultado		
Parámetros	De Entrada: <ul style="list-style-type: none">Ninguno De Salida: <ul style="list-style-type: none">Atributo consultado.		
Excepciones	Ninguna		
setAtributo			
Tipo	Método	Visibilidad	<input checked="" type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Protegida
Descripción	Actualiza el valor del atributo de la entidad deseado		
Parámetros	De Entrada: <ul style="list-style-type: none">atributo: valor nuevo del atributo De Salida: <ul style="list-style-type: none">Ninguno.		
Excepciones	Ninguna		

Tabla 211 métodos MessageRS.java

7.6.5. Diseño de jerarquías

El objetivo de esta tarea es revisar la jerarquía de clases que ha surgido en el modelo de clases a lo largo de las tareas anteriores.

La complejidad respecto a este aspecto es mínima en nuestro caso, en caso de herencia sólo hay un caso que se puede consultar en el punto 6.7.3. *Identificación de generalizaciones* del análisis.

7.6.6. Especificación de necesidades de migración y carga inicial de datos

Como ya se especificó en el punto 6.8.1. *Especificación de necesidades y de migración de datos y carga inicial* del análisis, el sistema no depende de una migración o carga inicial de datos.

7.7. Diseño físico de datos

En esta actividad se define la estructura física de datos que utilizará el sistema a partir del modelo de clases, de manera que teniendo presentes las características específicas del sistema de gestión de datos concreto a utilizar, los requisitos específicos para el sistema de información, y las particularidades del entorno tecnológico se consiga una mayor eficiencia en el tratamiento de los datos.

7.7.1. Diseño del modelo físico de datos

El objetivo de esta tarea es realizar el diseño del modelo físico de datos a partir del modelo de clases y el modelo lógico identificado en el análisis del sistema.

Como ya se ha ido mencionado a lo largo del proyecto el SGBD será MySQL, el cual es relacional. La aplicación interactuará con la base de datos a partir del subsistema modelo que utilizará JPA para ello, así de esta manera la aplicación podrá trabajar con objetos cuando en realidad por detrás la base de datos trabaja de forma relacional. MySQL proporciona el uso de estructuras como de disparadores, índices,... que pueden ayudar a la consistencia y rendimiento de la base de datos. Además se ha optado por este SGBD antes que otro como ORACLE debido a la baja complejidad del modelo físico, aunque no se descarta migrar a ese gestor en el caso de que en un futuro sea necesario.

Las clases entidades especificadas en los apartados anteriores serán las tablas de la base de datos del sistema. A continuación se adjunta el modelo físico de las entidades que será usado para la construcción de la base de datos.

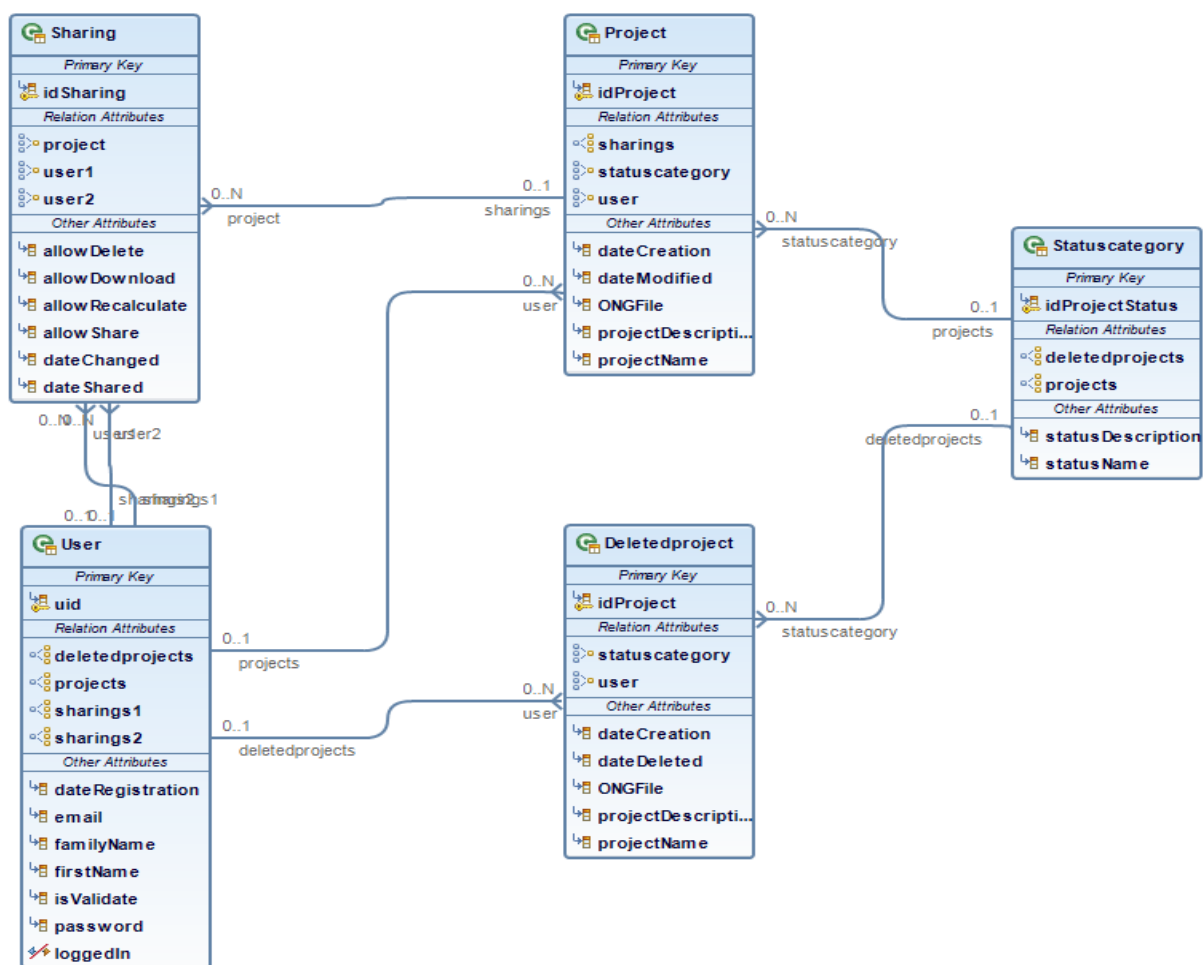


Ilustración 128 Modelo físico entidades

En base al anterior diagrama las tablas y columnas de la base de datos de la aplicación web quedarán de la siguiente forma:

Tabla	Descripción
Users	Almacena la información de los usuarios registrados en la aplicación
Projects	Almacena la información de los casos de simulación subidos al servidor de la aplicación
Deletedprojects	Almacena la información de los casos de simulación borrados en la aplicación
Statuscategories	Almacena la información de los estados posibles que puede tomar un caso de simulación
Sharings	Almacena la información de las comparticiones realizadas sobre los casos de simulación

Tabla 212 Tablas BD

Users		
Atributo	Descripción	Tipo
UID	ID del usuario	INT(11), NOT NULL, PK, AI
FirstName	Nombre del usuario	VARCHAR(60), NOT NULL
FamilyName	Apellido/s del usuario	VARCHAR(80), NOT NULL

Email	Correo electrónico del usuario	VARCHAR(60), NOT NULL, UQ
IsValidate	Marcador de activación de cuenta	TINYINT(1), NOT NULL, default '0'
DateRegistration	Fecha de creación de la cuenta	TIMESTAMP
password	Contraseña del usuario	VARCHAR(32), NOT NULL

Tabla 213 Tabla Users

Projects		
Atributo	Descripción	Tipo
IdProject	ID del caso de simulación	INT(11), NOT NULL, PK, AI
ProjectName	Nombre del caso de simulación	VARCHAR(60), NOT NULL
ProjectDescription	Descripción del caso de simulación	VARCHAR(150)
ONGFile	Ruta del caso de simulación en el servidor de la aplicación	BLOB, NOT NULL
UID	ID del usuario propietario	INT(11), NOT NULL, FK(Users)
IdProjectStatus	ID del estado del caso de simulación	TINYINT(4), NOT NULL, FK(Statuscategories), default '0'
DateCreation	Fecha en la cual el caso fue subido	TIMESTAMP, NOT NULL, default 'CURRENT_TIMESTAMP'
DateModifeied	Fecha de última simulación	TIMESTAMP, NOT NULL

Tabla 214 Tabla Projects

Deletedprojects		
Atributo	Descripción	Tipo
IdProject	ID del caso de simulación	INT(11), NOT NULL, PK
ProjectName	Nombre del caso de simulación	VARCHAR(60), NOT NULL
ProjectDescription	Descripción del caso de simulación	VARCHAR(150)
ONGFile	Ruta del caso de simulación en el servidor de la aplicación	BLOB, NOT NULL
UID	ID del usuario propietario	INT(11), NOT NULL, FK(Users)
IdProjectStatus	ID del estado del caso de simulación	TINYINT(4), NOT NULL, FK(Statuscategories)
DateCreation	Fecha en la cual el caso fue subido	TIMESTAMP, NOT NULL
DateDeleted	Fecha del borrado del caso	TIMESTAMP, NOT NULL

Tabla 215 Tabla Deletedprojects

Statuscategories		
Atributo	Descripción	Tipo
IdProjectStatus	ID del estado del caso de simulación	TINYINT(4), NOT NULL, PK
StatusName	Nombre del estado del caso	VARCHAR(45), NOT NULL
StatusDescription	Descripción del estado del caso	VARCHAR(150), NOT NULL

Tabla 216 Tabla Statuscategories

Sharings		
Atributo	Descripción	Tipo
IdSharing	ID de la compartición	INT(11), NOT NULL, PK, AI

IdProject	ID del caso de simulación	INT(11), NOT NULL, FK(Projects)
UID	ID del usuario al que se le ha compartido	INT(11), NOT NULL, FK(Users)
UIDsharer	ID del usuario que compartió	INT(11), NOT NULL, FK(Users)
DateShared	Fecha en la que se creó la compartición	TIMESTAMP, NOT NULL, default 'CURRENT_TIMESTAMP'
DateChanged	Fecha de última modificación de los permisos	TIMESTAMP, NOT NULL
AllowRecalculate	Permiso de volver a simular el caso	TINYINT(1), NOT NULL, default '0'
AllowDelete	Permiso de borrar caso	TINYINT(1), NOT NULL, default '0'
AllowDownload	Permiso de descargar caso	TINYINT(1), NOT NULL, default '0'
AllowShare	Permiso de compartir caso	TINYINT(1), NOT NULL, default '0'

Tabla 217 Tabla Sharings

7.7.2. Especificación de caminos de acceso a los datos y optimización del modelo físico

El objetivo de este punto es determinar los caminos de acceso a los datos persistentes en el sistema con el fin de optimizar el rendimiento y consumo de recursos, así como disminuir los tiempos de respuesta.

Esta información se puede consultar en el diagrama del apartado anterior y en el apartado del análisis relacionado con esta tarea.

7.7.3. Especificación de la distribución de datos

En este apartado se determina el modelo de distribución de datos, teniendo en cuenta la arquitectura del sistema que se ha especificado.

La base de datos se almacenará en el mismo servidor que la aplicación web y será accedida por el subsistema de la capa modelo de la aplicación web.

La capa modelo usará la tecnología JPA para realizar la tarea de comunicación con la base de datos. No se podrá acceder al contenido de la base de datos de otra manera por parte de los usuarios.

7.8. Verificación y aceptación de la arquitectura del sistema

El objetivo de esta actividad es garantizar la calidad de las especificaciones del diseño del sistema de información.

7.8.1. Verificación de las especificaciones de diseño

El objetivo de esta tarea es asegurar la calidad formal de los distintos modelos. Con esta comprobación se garantiza que se están siguiendo las pautas marcadas por el Project manager.

A continuación se muestra una lista con las comprobaciones realizadas hasta el este punto del proyecto. De esta manera se asegura que se cumplen los estándares y normas citados a lo largo de este documento.

- El diseño especificado es lo suficientemente claro y explicativo para que se pueda comenzar con la codificación de la aplicación sin problemas.
- Los subsistemas surgidos en la etapa de diseño recogen toda la funcionalidad que el análisis especifica en sus requisitos.
- No hay redundancia entre los sistemas ni ambigüedades en su explicación.
- Cada subsistema está dividido en componentes.
- Los componentes están correctamente comunicados.
- Hay coherencia entre el modelo de clases y el modelo físico.

7.8.2. Análisis de consistencia de las especificaciones de diseño

El objetivo de esta tarea es asegurar la coherencia de las especificaciones de diseño realizadas hasta el momento. Se debe comprobar que las especificaciones no son ambiguas y que no hay información duplicada. Esta consistencia se asegura entre especificaciones de diseño y con respecto a los modelos de análisis, estas comprobaciones se apoyarán en matrices de trazabilidad.

Las matrices de trazabilidad usadas para asegurar la consistencia son las siguientes, se mostrarán en el documento según el orden establecido en la siguiente lista:

- Clases de diseño/Casos de uso reales: Cada caso de uso real está al menos relacionado con una clase del diseño.
- Arquitectura del sistema/Entorno tecnológico del sistema de información: Cada nodo del particionamiento del sistema de información está soportado por el entorno tecnológico. También se da soporte a todas las necesidades de comunicaciones entre nodos.
- Modelo de clases/Modelo físico de datos: Los elementos del modelo físico de datos corresponden con los elementos utilizados por las clases identificadas, tanto de los subsistemas específicos como de soporte.
- Arquitectura del sistema/Requisitos de software: Verifica que hay consistencia entre el análisis y el diseño del sistema.

	CU-00	CU-01	CU-02	CU-03	CU-04	CU-05	CU-06	CU-07
<code>ServletController.java</code>	x	x	x	x	x	x	x	x
<code>AccountServlet.java</code>			x					

ActiveServlet.java	x							
DeleteServlet.java					x			
DownloadServlet.java					x			
LanguageServlet.java							x	
LoginServlet.java	x							
LogoutServlet.java		x						
RunServlet.java					x			
RegisterServlet.java	x							
PageServlet.java								x
RecoverServlet.java	x							
SeeServlet.java					x			
ShareServlet.java					x	x		
UploadServlet.java				x				
User.java	x	x	x		x	x		
Project.java	x	x		x	x	x		x
Sharing.java					x	x		
Deletedproject.java					x			
Statuscategory.java				x	x			
ProxyManager.java	x		x	x	x	x		x
Funciones.java	x	x		x	x	x	x	
Comunicacion.java								
HeartBeat.java								
Simulate.java				x	x			
MessageRQ.java				x	x			
MessageRS.java				x	x			

Tabla 218 Matriz Clases diseño/Casos de uso reales

	Vista	Controlador	Modelo
Servidor de la aplicación	x	x	x
Servidor/es de simulaciones			x

Tabla 219 Matriz arquitectura del sistema/Entorno tecnológico

	Users	Projects	Deletedprojects	Statuscategories	Sharings
ServletController.java	x				
AccountServlet.java	x				
ActiveServlet.java	x				
DeleteServlet.java		x	x		x
DownloadServlet.java		x			x
LanguageServlet.java					
LoginServlet.java	x				
LogoutServlet.java	x				
RunServlet.java		x		x	
RegisterServlet.java	x				
PageServlet.java		x			
RecoverServlet.java	x				
SeeServlet.java		x			x
ShareServlet.java		x			x
UploadServlet.java		x		x	
User.java	x				
Project.java		x			
Sharing.java					x
Deletedproject.java			x		
Statuscategory.java				x	
ProxyManager.java	x	x	x	x	x
Funciones.java		x			x

Comunicacion.java					
HeartBeat.java					
Simulate.java		X		X	
MessageRQ.java		X		X	
MessageRS.java		X		X	

Tabla 220 Matriz clases/datos

Vista															
	index.jsp	signup.jsp	forgotten.jsp	account.jsp	activate.jsp	inicio.jsp	lenguaje.jsp	recover.jsp	recuperar.jsp	registro.jsp	see.jsp	share.jsp	top.jsp	upload.jsp	error.jsp
SR-F00	X														X
SR-F01		X								X					
SR-F02			X					X	X						
SR-F03					X										
SR-F04				X									X		
SR-F05						X									
SR-F06						X									
SR-F07	X												X		
SR-F08						X					X	X			
SR-F09						X									
SR-F10						X									
SR-F11												X			
SR-F12							X						X		
SR-F13															
SR-F14													X		
SR-F15						X									
SR-F16														X	
SR-F17															
SR-F18														X	
SR-F19															
SR-F20						X									
SR-F21											X				
	index.jsp	signup.jsp	forgotten.jsp	account.jsp	activate.jsp	inicio.jsp	lenguaje.jsp	recover.jsp	recuperar.jsp	registro.jsp	see.jsp	share.jsp	top.jsp	upload.jsp	error.jsp
SR-S00	X			X				X							
SR-S01															
SR-S02															
SR-S03						X						X			
SR-M00	X														
SR-M01												X			
SR-M02			X												
SR-M03	X														X
SR-M04														X	
SR-M05						X									
SR-M06															
SR-R00															
SR-R01															
SR-R02															
SR-R03															
SR-R04															
SR-I00															
SR-I01															
SR-I02															
SR-I03															
SR-O00								X							

SR-O01					X										
SR-O02															
SR-O03															
SR-O04															
SR-O05															
SR-O06															
SR-O07															
SR-O08															

Tabla 221 Matriz Vista/Requisitos Software

Controlador															
	ServletController.java	AccountServlet.java	ActivateServlet.java	DeleteServlet.java	DownloadServlet.java	LenguajeServlet.java	LoginServlet.java	LogoutServlet.java	RunServlet.java	RegisterServlet.java	PageServlet.java	RecoverServlet.java	SeeServlet.java	ShareServlet.java	UploadServlet.java
SR-F00	X						X								
SR-F01	X									X					
SR-F02	X											X			
SR-F03	X		X												
SR-F04	X	X													
SR-F05	X						X								
SR-F06	X						X								
SR-F07	X							X							
SR-F08	X			X	X				X				X	X	
SR-F09															
SR-F10	X										X				
SR-F11	X													X	
SR-F12	X					X									
SR-F13	X						X								
SR-F14															
SR-F15															
SR-F16	X														X
SR-F17	X														X
SR-F18															
SR-F19															
SR-F20	X			X											
SR-F21	X												X		
	ServletController.java	AccountServlet.java	ActivateServlet.java	DeleteServlet.java	DownloadServlet.java	LenguajeServlet.java	LoginServlet.java	LogoutServlet.java	RunServlet.java	RegisterServlet.java	PageServlet.java	RecoverServlet.java	SeeServlet.java	ShareServlet.java	UploadServlet.java
SR-S00															
SR-S01	X									X					
SR-S02	X									X					X
SR-S03	X			X	X				X				X	X	
SR-M00	X						X								
SR-M01	X													X	
SR-M02	X											X			
SR-M03	X														

SR-M04	X														X
SR-M05															
SR-M06															
SR-R00															
SR-R01															
SR-R02															
SR-R03															
SR-R04															
SR-I00	X		X						X		X				
SR-I01															
SR-I02															
SR-I03	X														
SR-O00	X										X				
SR-O01	X		X						X						
SR-O02	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
SR-O03	X														
SR-O04	X														X
SR-O05	X						X	X							
SR-O06	X														
SR-O07	X								X						
SR-O08	X								X						

Tabla 222 Matriz Controlador/Requisitos Software

	Modelo											
	Use r.java	Proje ct.java	Sharin g.java	Deletedproj ect.java	Statusca tegory.j ava	ProxyMana ger.java	Funcione s.java	Comunicaci on.java	HeartBe at.java	Simulat e.java	MessageR Q.java	Message RS.java
SR-F00	X					X						
SR-F01	X					X						
SR-F02	X					X						
SR-F03	X					X						
SR-F04	X					X						
SR-F05	X	X				X	X					
SR-F06	X	X				X	X					
SR-F07	X	X				X	X					
SR-F08		X			X	X				X	X	X
SR-F09												
SR-F10												
SR-F11	X	X	X			X						
SR-F12												
SR-F13												
SR-F14												
SR-F15		X			X	X						
SR-F16		X			X	X						
SR-F17												

SR-F18		X				X						
SR-F19		X			X	X				X	X	X
SR-F20		X		X		X						
SR-F21												
	Use r.java	Proyec t.java	Sharin g.java	Deletedproj ect.java	Statusca tegory.j ava	ProxyMana ger.java	Funcione s.java	Comunicaci on.java	HeartBe at.java	Simulat e.java	MessageR Q.java	Message RS.java
SR-S00												
SR-S01	X					X	X					
SR-S02	X	X	X			X						
SR-S03	X		X			X	X					
SR-M00	X					X	X					
SR-M01	X		X			X						
SR-M02	X					X						
SR-M03												
SR-M04							X					
SR-M05												
SR-M06										X	X	X
SR-R00												
SR-R01												
SR-R02												
SR-R03							X	X	X			
SR-R04		X					X			X	X	X
SR-I00	X	X					X	X	X	X	X	X
SR-I01								X	X	X	X	X
SR-I02						X						
SR-I03												
SR-O00						X	X					
SR-O01						X	X					
SR-O02												
SR-O03												
SR-O04		X				X						
SR-O05												
SR-O06								X	X			
SR-O07		X				X	X			X	X	X
SR-O08		X				X				X	X	X

Tabla 223 Matriz Modelo/Requisitos Software

7.9. Especificación técnica del plan de pruebas

El plan de pruebas a desarrollar permitirá detectar errores y corregirlos, comprobar que la aplicación cuenta con las funcionalidades requeridas y que estas realizan lo especificado, etc.

De esta manera se obtendrá un sistema robusto y consistente. El plan de pruebas se desarrollará en su completitud en el apartado 8. *Plan de validación y verificación del software*.

8. Plan de validación y verificación del software

Este apartado del documento recoge las pruebas que se realizarán al sistema con el fin de comprobar que se cumplen los requisitos establecidos y descubrir defectos que pueda tener en su comportamiento.

8.1. Objetivo

El principal objetivo es elaborar un completo plan de pruebas de la aplicación web que permita comprobar que el sistema funciona correctamente y que cumple lo especificado a lo largo de la memoria.

Esta tarea verificará y validará el sistema de cara al cliente. Se comprobará que se cumplen los requisitos, especificaciones, que no hay defectos...

8.2. Especificación del plan de pruebas

En esta actividad se establece la definición del plan de pruebas. El cual debe ser claro y conciso, para que a la hora de realizar dichas pruebas no hay confusiones.

8.2.1. Definición del alcance de pruebas

En función del sistema desarrollado algunos tipos de pruebas son más importantes que otras. De acuerdo al sistema que se ha desarrollado en este TFG los tipos de pruebas que se han considerado a realizar son:

- **Pruebas unitarias:** Estas pruebas tienen como objetivo comprobar de forma aislada que cada uno de los componentes del sistema y los métodos de las distintas clases de implementación funcionan correctamente. La principal ventaja de este tipo de pruebas está en que facilita la detección de errores, al tener verificados los distintos métodos y componentes de forma separada.
- **Pruebas de aceptación:** Estas pruebas tienen como objetivo comprobar que todas las necesidades del usuario han sido satisfechas, teniendo en cuenta la especificación de requisitos de usuario.
- **Pruebas del sistema:** Estas pruebas tienen como objetivo ejercitar profundamente el sistema comprobando la integración del sistema de información globalmente, verificando el funcionamiento correcto de las interfaces entre los distintos subsistemas que lo componen y con el resto de sistemas de información con los que se comunica.
- **Pruebas de implantación:** Estas pruebas tienen como objetivo comprobar el funcionamiento correcto del sistema integrado de hardware y software en el entorno de operación, y permitir al usuario que, desde el punto de vista de operación, realice la

aceptación del sistema una vez instalado en su entorno real y en base al cumplimiento de los requisitos no funcionales especificados.

- **Pruebas de integración:** Estas pruebas tienen como objetivo verificar el correcto ensamblaje entre los distintos componentes una vez que han sido probados unitariamente con el fin de comprobar que interactúan correctamente a través de sus interfaces, tanto internas como externas, cubren la funcionalidad establecida y se ajustan a los requisitos no funcionales especificados en las verificaciones correspondientes.

El entorno donde puedan ser ejecutadas estas pruebas será el mismo que se ha especificado en 7.3.5. *Especificación del entorno tecnológico*. De esta manera se asegura que la ejecución de estas se realice de una manera correcta y satisfactoria.

Algunas principales características que deben tener las pruebas que se van a especificar son las siguientes:

- **Completas:** las pruebas deben cubrir la totalidad del sistema.
- **Reutilizables:** las pruebas deben ser lo más genéricas para poder reutilizarlas en otros casos.
- **Independientes:** la ejecución de una prueba no debe depender del resultado de la ejecución de otra.
- **Automatizables:** se debe eliminar la ejecución manual de las pruebas cuando sea posible.

Todas las pruebas que se especificarán a continuación estarán recogidas en una tabla con unos campos específicos dependiendo del tipo de prueba.

8.2.2. Pruebas unitarias

El objetivo de estas pruebas es probar el correcto funcionamiento de cada componente del sistema por separado. La estructura de las tablas que contendrán la especificación de estas pruebas es la siguiente:

- **Identificador:** identifica unívocamente la prueba. Su estructura será de PY-XX, donde Y es el tipo de prueba y XX el número. Para este tipo de pruebas Y tomará el valor de U.
- **Objetivo:** breve descripción de lo que se pretende conseguir con la prueba.
- **Subsistema:** nombre del subsistema afectado por la prueba.
- **Clase:** nombre de la clase testeada.
- **Método:** nombre del método evaluado.
- **Entrada:** valores de entrada de la prueba.
- **Salida:** valores de salida generada.
- **Precondición:** estado del sistema apto para realizar la prueba.
- **Postcondición:** estado del sistema tras realizar la prueba.
- **Requisito/s:** requisito software relacionado con la prueba.

PU-00			
Objetivo	Comprobar que el panel de login carga bien	Subsistema	Vista
Clase	index.jsp	Método	-
Entrada	-	Salida	Muestra panel de login
Precondición	Entrar a la página sin estar logueado		
Postcondición	Se muestra correctamente la cajita para loguearse		
Requisito/s	SR-F00		

Tabla 224 PU-00

PU-01			
Objetivo	Comprobar que el repositorio se muestra bien	Subsistema	Vista
Clase	inicio.jsp	Método	-
Entrada	Response de LoginServlet	Salida	Muestra repositorio
Precondición	Que el login sea correcto		
Postcondición	Se muestra correctamente el repositorio del usuario		
Requisito/s	SR-F05, SR-F09, SR-F10		

Tabla 225 PU-01

PU-02			
Objetivo	Comprobar que el panel de cuenta muestra los datos bien	Subsistema	Vista
Clase	account.jsp	Método	-
Entrada	Response de AccountServlet	Salida	Muestra datos de registro
Precondición	Usuario logueado y que haya seleccionado la opción e 'Mi cuenta'		
Postcondición	Se muestran todos los datos de registro del usuario		
Requisito/s	SR-F04		

Tabla 226 PU-02

PU-03			
Objetivo	Comprobar que el panel de compartición de un caso muestra correctamente la información	Subsistema	Vista
Clase	shared.jsp	Método	-
Entrada	Response de ShareServlet	Salida	Muestra panel de compartir
Precondición	Que el login sea correcto y el usuario elija la opción compartir de un caso simulado		
Postcondición	Se muestra correctamente la información de compartición de ese caso		
Requisito/s	SR-F11		

Tabla 227 PU-03

PU-04			
Objetivo	Comprobar que el panel de subida de casos se muestra correctamente	Subsistema	Vista
Clase	upload.jsp	Método	-
Entrada	-	Salida	Muestra panel de subida
Precondición	Que el login sea correcto y el usuario elija la opción añadir caso		
Postcondición	Se muestra correctamente el panel para subir un nuevo caso		

Requisito/s	SR-F16
--------------------	--------

Tabla 228 PU-04

PU-05			
Objetivo	Comprobar que el cambio de idioma se refleja correctamente	Subsistema	Vista
Clase	*.jsp	Método	-
Entrada	Response de LanguageServlet	Salida	Muestra página con idioma cambiado
Precondición	Que el usuario elija un idioma distinto del actual		
Postcondición	Se muestra correctamente la página actual con el nuevo idioma		
Requisito/s	SR-F12		

Tabla 229 PU-05

PU-06			
Objetivo	Comprobar que el cambio de página en el repositorio se muestra bien	Subsistema	Vista
Clase	inico.jsp	Método	-
Entrada	Response de PageServlet	Salida	Muestra página siguiente/anterior del repositorio
Precondición	Que el usuario tenga más de cuatro casos propios o compartidos en el servidor y elija la opción de cambiar página		
Postcondición	Se muestra la página siguiente/anterior con el resto de casos		
Requisito/s	SR-F10		

Tabla 230 PU-06

PU-07			
Objetivo	Comprobar que el login se realiza correctamente	Subsistema	Controlador
Clase	LoginServlet.java	Método	doPost()
Entrada	Request y response	Salida	Response con los datos para la carga del repositorio
Precondición	Que el login sea correcto		
Postcondición	El sistema devuelve todos los datos de los casos del usuario y descomprime estos en la carpeta temporal del usuario		
Requisito/s	SR-F06		

Tabla 231 PU-07

PU-08			
Objetivo	Comprobar que el logout se realiza correctamente	Subsistema	Controlador
Clase	LogoutServlet.java	Método	doGet()
Entrada	Request y response	Salida	Redirección a página de login
Precondición	Que el usuario este logueado y elija la opción 'logout'		
Postcondición	El sistema borra la carpeta temporal del usuario y redirige a la página de login		
Requisito/s	SR-F07		

Tabla 232 PU-08

PU-09			
Objetivo	Comprobar que la muestra de datos del usuario se realiza bien	Subsistema	Controlador
Clase	AccountServlet.java	Método	doGet()
Entrada	Request y response	Salida	Redirección a página de account
Precondición	Que el usuario este logueado y elija la opción 'Mi cuenta'		
Postcondición	El sistema carga la información de la cuenta del usuario en la página jsp		
Requisito/s	SR-F04		

Tabla 233 PU-09

PU-10			
Objetivo	Comprobar que la subida de un caso es correcta	Subsistema	Controlador
Clase	UploadServlet.java	Método	doPost()
Entrada	Request y response	Salida	Redirección a página de repositorio
Precondición	Que el usuario este logueado y suba un caso		
Postcondición	El sistema comprueba que el caso es correcto y lo almacena en el servidor		
Requisito/s	SR-F16, SR-F17, SR-F18, SR-F19		

Tabla 234 PU-10

PU-11			
Objetivo	Comprobar que el cambio de permisos de una compartición es correcto	Subsistema	Controlador
Clase	ShareServlet.java	Método	doPost()
Entrada	Request y response	Salida	Redirección a página de compartición
Precondición	Que el usuario este logueado y cambie los permisos de un usuario en una compartición		
Postcondición	El sistema actualiza la información de esa compartición y redirige al panel de compartición de ese caso		
Requisito/s	SR-F11		

Tabla 235 PU-11

PU-12			
Objetivo	Comprobar que se guarda correctamente el registro de un nuevo usuario	Subsistema	Modelo
Clase	ProxyManager.java	Método	createObject()
Entrada	Objeto User con los datos del nuevo usuario	Salida	Retorna el objeto insertado
Precondición	Un usuario debe registrarse y la base de datos debe estar accesible		
Postcondición	El sistema almacena en la base de datos la información del usuario		
Requisito/s	SR-F01, SR-F03		

Tabla 236 PU-12

PU-13			
-------	--	--	--

Objetivo	Comprobar que se guarda correctamente la modificación de los datos de un usuario	Subsistema	Modelo
Clase	ProxyManager.java	Método	updateObject()
Entrada	Objeto User con los valores nuevos de los datos del usuario	Salida	Retorna el objeto actualizado
Precondición	Un usuario debe actualizar sus datos y la base de datos debe estar accesible		
Postcondición	El sistema actualiza en la base de datos la información del usuario		
Requisito/s	SR-F04		

Tabla 237 PU-13

PU-12			
Objetivo	Comprobar que se guarda correctamente la subida de un nuevo caso	Subsistema	Modelo
Clase	ProxyManager.java	Método	createObject()
Entrada	Objeto Project con los datos del nuevo caso	Salida	Retorna el objeto insertado
Precondición	Un usuario debe subir un nuevo caso y la base de datos debe estar accesible		
Postcondición	El sistema almacena en la base de datos la información del caso		
Requisito/s	SR-F16		

Tabla 238 PU-14

PU-15			
Objetivo	Comprobar que se guarda correctamente la creación de una compartición	Subsistema	Modelo
Clase	ProxyManager.java	Método	createObject()
Entrada	Objeto Sharing con los datos de la nueva compartición	Salida	Retorna el objeto insertado
Precondición	Un usuario debe compartir un caso con un usuario existente y la base de datos debe estar accesible		
Postcondición	El sistema almacena en la base de datos la información de la compartición		
Requisito/s	SR-F08		

Tabla 239 PU-15

PU-16			
Objetivo	Comprobar que se actualiza correctamente una compartición	Subsistema	Modelo
Clase	ProxyManager.java	Método	updateObject()
Entrada	Objeto Sharing con los datos modificados de la compartición	Salida	Retorna el objeto actualizado
Precondición	Un usuario debe actualizar los permisos de un usuario en una compartición y la base de datos debe estar accesible		
Postcondición	El sistema actualiza en la base de datos la información de la compartición		
Requisito/s	SR-F11		

Tabla 240 PU-16

PU-17			
-------	--	--	--

Objetivo	Comprobar que se borra correctamente un caso	Subsistema	Modelo
Clase	ProxyManager.java	Método	deleteObject()
Entrada	Objeto Project con los datos del caso a borrar	Salida	String verificando que todo fue correcto
Precondición	Un usuario debe borrar un caso ya simulado o con errores y la base de datos debe estar accesible		
Postcondición	El sistema elimina ese registro de la base de datos		
Requisito/s	SR-F20		

Tabla 241 PU-20

8.2.3. Pruebas de aceptación

El objetivo de estas pruebas es validar que la aplicación web cumple con los requisitos de usuario. Las pruebas serán especificadas en una tabla con los siguientes campos:

- *Identificador*: identifica unívocamente la prueba. Su estructura será de PY-XX, donde Y es el tipo de prueba y XX el número. Para este tipo de pruebas Y tomará el valor de A.
- *Objetivo*: breve descripción de lo que se pretende conseguir con la prueba.
- *Entrada*: valores de entrada de la prueba.
- *Salida*: valores de salida generada.
- *Precondición*: estado del sistema apto para realizar la prueba.
- *Postcondición*: estado del sistema tras realizar la prueba.
- *Requisito/s*: Requisito de usuario con el que está relacionado

PA-00			
Objetivo	Comprobar que al introducir unos datos de acceso correctos en el panel de login nos redirige al repositorio de ese usuario		
Entrada	Email y contraseña correctas	Salida	Panel de repositorio con la lista de casos asociada a ese usuario
Precondición	El usuario debe introducir su email y contraseña de registro en la ventana de login		
Postcondición	Se debe redirigir al repositorio de este usuario y mostrar sus casos		
Requisito/s	UR-C00, UR-C04, UR-C06, UR-C07, UR-R11		

Tabla 242 PA-00

PA-01			
Objetivo	Comprobar que al introducir el email para recuperar la contraseña llega un correo a esa dirección con los pasos a seguir		
Entrada	Email de registro	Salida	Mensaje notificando que se envió un correo a esa dirección
Precondición	El usuario debe introducir su email en el apartado de recuperación de cuenta		
Postcondición	El sistema debe enviar un correo con los pasos a esa dirección		
Requisito/s	UR-C01		

Tabla 243 PA-01

PA-02			
-------	--	--	--

Objetivo	Comprobar que se crea un nuevo usuario tras completar el registro		
Entrada	Se rellenan todos los campos de la pantalla de registro y se envía	Salida	Mensaje notificando que se envió un correo de activación al email de registro
Precondición	El usuario debe completar correctamente todos los campos de registro		
Postcondición	El sistema debe enviar un correo al email de registro con los pasos para activar la cuenta		
Requisito/s	UR-C02, UR-C03		

Tabla 244 PA-02

PA-03			
Objetivo	Comprobar que al compartir un caso este aparece en el repositorio del compartido		
Entrada	Se comparte un caso ya simulado con un usuario registrado en la aplicación distinto al propietario de él	Salida	Aparece este nuevo usuario en el panel de compartición del que compartió y en el repositorio del usuario compartido aparece el caso
Precondición	El usuario debe introducir el email del usuario a compartir en la sección de compartir casos		
Postcondición	El sistema da acceso al caso al nuevo usuario con los permisos que el usuario que le compartió le asigne		
Requisito/s	UR-C09, UR-R05		

Tabla 245 PA-03

PA-04			
Objetivo	Comprobar que la subida de un caso al servidor es correcta		
Entrada	Se introduce un caso de simulación en el área de subidas	Salida	Tras completarse la subida el nuevo caso aparece en el repositorio del usuario
Precondición	El usuario debe subir un caso		
Postcondición	El sistema debe guardar el caso en el servidor y su información en la BD. Después lanzará su simulación		
Requisito/s	UR-C10, UR-C06		

Tabla 246 PA-04

PA-05			
Objetivo	Comprobar que se puede ver la solución de un caso simulado		
Entrada	El usuario pulsa el icono de ver la solución de un caso simulado	Salida	Se muestra una pantalla con la solución de este caso
Precondición	El usuario debe elegir la opción de ver solución de un caso con la simulación finalizada		
Postcondición	El sistema redirige a una página en la que se muestra el contenido del HTML del caso		
Requisito/s	UR-C11		

Tabla 247 PA-05

PA-06			
-------	--	--	--

Objetivo	Comprobar que se puede volver a simular un caso		
Entrada	El usuario pulsa el botón de simular de un caso simulado	Salida	El caso se vuelve a simular
Precondición	El usuario debe volver a lanzar la simulación de un caso		
Postcondición	El sistema manda el aviso al servidor de simulaciones		
Requisito/s	UR-C12		

Tabla 248 PA-06

PA-07			
Objetivo	Comprobar que se puede descargar un caso ya simulado		
Entrada	El usuario pulsa el botón de descargar de un caso simulado	Salida	El caso descarga una copia para el usuario
Precondición	El usuario debe elegir la opción de descargar un caso		
Postcondición	El sistema lanza una descarga para guardar una copia localmente en la máquina del usuario		
Requisito/s	UR-C13		

Tabla 249 PA-07

PA-08			
Objetivo	Comprobar que se puede borrar un caso ya simulado		
Entrada	El usuario pulsa el botón de borrar de un caso simulado	Salida	Se pide confirmación y si es afirmativa borra el caso
Precondición	El usuario debe elegir la opción de borrar un caso		
Postcondición	El sistema pide confirmación y si es afirmativa borra el caso del servidor y mueve su registro en la base de datos a la tabla de casos borrados		
Requisito/s	UR-C14, UR-C15		

Tabla 250 PA-08

PA-09			
Objetivo	Comprobar que el idioma cambia tras elegir otro		
Entrada	El usuario elige un idioma distinto al actual en la aplicación	Salida	Se recarga la página actual con el idioma cambiado
Precondición	El usuario debe elegir un idioma distinto del actual		
Postcondición	El sistema recarga la página actual con la traducción y pone ese idioma como el actual para ese usuario		
Requisito/s	UR-C16, UR-R10		

Tabla 251 PA-09

PA-10			
Objetivo	Comprobar que si se tiene más de cuatro casos propios o compartidos en el servidor se permite cambiar de página en el repositorio		
Entrada	El usuario cuenta con más de cuatro casos propios o compartidos subidos y cambia de página del repositorio	Salida	Se muestra la siguiente página del repositorio con los casos que antes no se mostraban
Precondición	El usuario debe elegir la opción de página siguiente/anterior		

Postcondición	El sistema carga la información de los nuevos casos a mostrar y redirige al usuario a la nueva página
Requisito/s	UR-R18

Tabla 252 PA-10

8.2.4. Pruebas de sistema

El objetivo de estas pruebas es valorar el comportamiento del sistema en su totalidad. La definición de estas pruebas en sus tablas tendrán los siguientes campos:

- *Identificador*: identifica unívocamente la prueba. Su estructura será de PY-XX, donde Y es el tipo de prueba y XX el número. Para este tipo de pruebas Y tomará el valor de S.
- *Objetivo*: breve descripción de lo que se pretende conseguir con la prueba.
- *Entrada*: valores de entrada de la prueba.
- *Salida*: valores de salida generada.
- *Aceptación*: valora si la prueba es correcta o no.
- *Requisito/s*: requisito software con el que está relacionado

PS-00			
Objetivo	Verificar la disponibilidad del sistema en cualquier momento	Aceptación	<input checked="" type="checkbox"/> Satisfactoria <input type="checkbox"/> No satisfactoria
Entrada	Se realizan peticiones al servidor de la aplicación cada 5 min durante un periodo de 2 horas	Salida	El sistema realiza correctamente las peticiones y responde sin demora
Requisito/s	SR-C00		

Tabla 253 PS-00

PS-01			
Objetivo	Verificar que el sistema soporta un gran número de usuarios	Aceptación	<input checked="" type="checkbox"/> Satisfactoria <input type="checkbox"/> No satisfactoria
Entrada	Se realizan pruebas con 50 usuarios logueados y activos en la aplicación realizando peticiones continuamente durante 5 minutos	Salida	El sistema realiza correctamente las peticiones y responde sin fallos durante este tiempo
Requisito/s	SR-C00		

Tabla 254 PS-01

PS-02			
Objetivo	Verificar que todas las vistas de la aplicación se muestran correctamente	Aceptación	<input checked="" type="checkbox"/> Satisfactoria <input type="checkbox"/> No satisfactoria
Entrada	El usuario navega por todas las secciones de la aplicación	Salida	El sistema muestra correctamente el contenido de cada una sin fallos

8.2.5. Pruebas de implantación

El objetivo de estas pruebas es asegurar la ausencia de errores en el entorno de operación en el que se ejecute. Las tablas que contendrán estas pruebas tendrán los siguientes campos:

- **Identificador:** identifica unívocamente la prueba. Su estructura será de PY-XX, donde Y es el tipo de prueba y XX el número. Para este tipo de pruebas Y tomará el valor de I.
- **Objetivo:** breve descripción de lo que se pretende conseguir con la prueba.
- **Entrada:** valores de entrada de la prueba.
- **Salida:** valores de salida generada.
- **Especificación:** su función es identificar a que especificación responde el resultado. Los valores que pueden tomar son:
 - Recuperación: si se produce un fallo se comprueba que el sistema se recupera correctamente.
 - Seguridad: se comprueban los mecanismos de seguridad implementados, para ver si realmente cumplen con su función.
 - Rendimiento: se analiza el consumo de recursos y el tiempo de espera de respuesta del sistema para así comprobar que es adecuado.
 - Comunicaciones: comprobar que la comunicación entre las distintas partes del sistema es correcta.
- **Aceptación:** valora si la prueba es correcta o no.
- **Requisito/s:** requisito software con el que está relacionado.

PI-00			
Objetivo	Comprobar que el tiempo medio de carga de login/logout no supera lo especificado	Aceptación	<input checked="" type="checkbox"/> Satisfactoria <input type="checkbox"/> No satisfactoria
Entrada	Realizar login y logout con un usuario con un número considerable de casos subidos	Salida	La medida de los tiempos no puede superar los máximos establecidos en ninguna de estas dos tareas
Especificación	Rendimiento		
Requisito/s	SR-R00, SR-R01		

Tabla 256 PI-00

PI-01			
Objetivo	Comprobar que el tiempo medio de carga de una página normal no supera lo especificado	Aceptación	<input checked="" type="checkbox"/> Satisfactoria <input type="checkbox"/> No satisfactoria

Entrada	Realizar navegación entre las páginas de la aplicación	Salida	La medida de los tiempos no puede superar los máximos establecidos
Especificación	Rendimiento		
Requisito/s	SR-R02		

Tabla 257 PI-01

PI-02			
Objetivo	Comprobar que las contraseñas se guardan cifradas en la base de datos	Aceptación	<input checked="" type="checkbox"/> Satisfactoria <input type="checkbox"/> No satisfactoria
Entrada	Realizar el registro de un nuevo usuario	Salida	El sistema debe almacenar en la base de datos el registro de este usuario con su contraseña cifrada en MD5
Especificación	Seguridad		
Requisito/s	SR-S01		

Tabla 258 PI-02

8.2.6. Pruebas de integración

Estas pruebas tienen como el objetivo probar la correcta verificación de los grupos de componentes definidos. Este tipo de pruebas se hará teniendo en cuenta los casos de uso definidos en el diseño. La tabla que contendrá su especificación tendrá los siguientes campos:

- *Identificador*: identifica unívocamente la prueba. Su estructura será de PY-XX, donde Y es el tipo de prueba y XX el número. Para este tipo de pruebas Y tomará el valor de N.
- *Objetivo*: breve descripción de lo que se pretende conseguir con la prueba.
- *Entrada*: valores de entrada de la prueba.
- *Salida*: valores de salida generada.
- *Procedimiento*: describe el procedimiento a seguir para realizar la prueba correctamente.
- *Componentes*: componentes que intervienen en la prueba.
- *Precondición*: estado del sistema apto para realizar la prueba.
- *Postcondición*: estado del sistema tras realizar la prueba.
- *Caso de uso*: caso de uso relacionado con la prueba

PN-00			
Objetivo	Loguearse en la aplicación		Componentes JSP pages, Servlets, JPA, Base de datos
Entrada	Email y contraseña de usuario	Salida	Carga de casos asociados a ese usuario
Procedimiento	<ul style="list-style-type: none"> • Entrar a la página • Introducir email y contraseña en el formulario y dar en login. 		

	<ul style="list-style-type: none"> El sistema comprueba si son correctos los datos y empieza a cargar los casos. Esperar redirección al repositorio
Precondición	El usuario debe estar registrado y acceder a la aplicación
Postcondición	La página debe mostrar la lista de casos de ese usuario
Caso de uso	CU-00_1

Tabla 259 PN-00

PN-01			
Objetivo	Recuperar contraseña		Componentes JSP pages, Servlets, JPA, Base de datos
Entrada	a. Email de registro b. Nueva contraseña	Salida	a. Se envía correo a email con los pasos a seguir b. Se actualiza la contraseña
Procedimiento	<ul style="list-style-type: none"> Entrar a la página y elegir la opción de recuperar contraseña Introducir email de registro. El sistema comprueba si existe y manda correo. Visitar el link del correo y poner nueva contraseña. 		
Precondición	El usuario debe estar registrado y acceder a la sección de recuperación de contraseña.		
Postcondición	El sistema debe enviar correo con los pasos y actualizar posteriormente la contraseña.		
Caso de uso	CU-00_2		

Tabla 260 PN-01

PN-02			
Objetivo	Registrarse en la aplicación		Componentes JSP pages, Servlets, JPA, Base de datos
Entrada	a. Email, contraseña, nombre y apellidos b. Link de activación	Salida	a. Se registra el nuevo usuario en el sistema. b. Se activa la cuenta del usuario
Procedimiento	<ul style="list-style-type: none"> Entrar a la página y elegir la opción de registro Introducir datos de registro. El sistema comprueba la validez de estos y crea el registro del nuevo usuario en la base de datos. Visitar el link del correo para activar la cuenta. El sistema marca la cuenta del usuario como activada 		
Precondición	El usuario debe estar no registrado y acceder a la sección de registro		
Postcondición	El sistema debe registrar al nuevo usuario en la base de datos y posteriormente activar su cuenta cuando se visite el link		
Caso de uso	CU-00_3		

Tabla 261 PN-02

PN-03			
Objetivo	Desconectarse de la aplicación	Componentes	JSP pages, Servlets

Entrada	Petición de logout	Salida	Se eliminan los archivos descomprimidos de los casos del usuario, se invalida la sesión y se redirige a la página de inicio
Procedimiento	<ul style="list-style-type: none"> • Una vez logueado elegir la opción de logout • El sistema busca la carpeta temporal del usuario y la borra. • El sistema invalida la sesión y redirige a la página de inicio. 		
Precondición	El usuario debe estar logueado y elegir la opción de logout		
Postcondición	El sistema debe borrar los archivos temporales del usuario, invalidar su sesión y redirigirle al inicio de la aplicación		
Caso de uso	CU-01		

Tabla 262 PN-03

PN-04			
Objetivo	Cambiar los datos de registro	Componentes	JSP pages, Servlets, JPA, Base de datos
Entrada	Nombre, apellidos o contraseña cambiados	Salida	Se actualizan en la base de datos los campos cambiados
Procedimiento	<ul style="list-style-type: none"> • Una vez logueado elegir la opción de mi cuenta • Cambiar los atributos deseados y enviar • El sistema comprueba que son correctos y actualiza en la base de datos 		
Precondición	El usuario debe estar logueado y elegir la opción de mi cuenta		
Postcondición	El sistema debe actualizar en la base de datos los nuevos valores y el usuario debe ver el cambio en la aplicación		
Caso de uso	CU-02		

Tabla 263 PN-04

PN-05			
Objetivo	Subir caso de simulación	Componentes	JSP pages, Servlets, JPA, Base de datos
Entrada	Archivo ongf, zip o rar que cumple los estándares	Salida	Se almacena en el servidor el archivo y su información en la base de datos
Procedimiento	<ul style="list-style-type: none"> • Una vez logueado elegir la opción añadir caso y subir uno/s • El sistema comprueba que el/los archivo/s son correctos. • El sistema almacena la información del caso en la base de datos y una copia de este en el servidor. • El sistema lanza la simulación de los casos subidos. • El sistema redirige a la página del repositorio 		
Precondición	El usuario debe estar logueado y subir casos		
Postcondición	El sistema debe comprobar que son válidos, guardarlos en el servidor, en la base de datos y lanzar su simulación		
Caso de uso	CU-03		

Tabla 264 PN-05

PN-06

Objetivo	Visualizar solución del caso		Componentes	JSP pages, Servlets
Entrada	Petición de visionado de solución del caso	Salida	El sistema busca la carpeta temporal de ese caso en el servidor y muestra su contenido HTML en una página	
Procedimiento	<ul style="list-style-type: none">• Una vez logueado elegir la opción de ver solución de un caso simulado.• El sistema busca en el servidor la carpeta temporal de ese caso• El sistema vuelca el HTML de ese caso en una página• Se muestra esta página al usuario			
Precondición	El usuario debe estar logueado y elegir la opción de ver solución de un caso ya simulado			
Postcondición	El sistema debe mostrar el contenido HTML donde está la solución del caso al usuario			
Caso de uso	CU-04_1			

Tabla 265 PN-06

PN-07				
Objetivo	Re-lanzar simulación del caso		Componentes	JSP pages, Servlets, JPA, Base de datos
Entrada	Petición de iniciar simulación del caso	Salida	El sistema manda la petición al servidor de simulaciones y establece una comunicación continua con él hasta que finaliza la simulación	
Procedimiento	<ul style="list-style-type: none">• Una vez logueado elegir la opción de simular de un caso simulado.• El sistema manda la petición al servidor de simulaciones• El sistema va actualizando el estado del caso por cada respuesta que le llega del servidor de simulaciones• Al finalizar la simulación el sistema descomprime el HTML del caso y actualiza el estado de este a 'simulado'			
Precondición	El usuario debe estar logueado y elegir la opción de simular de un caso ya simulado			
Postcondición	El sistema debe gestionar la simulación con el servidor de simulaciones			
Caso de uso	CU-04_2			

Tabla 266 PN-07

PN-08				
Objetivo	Compartir un caso		Componentes	JSP pages, Servlets, JPA, Base de datos
Entrada	Email de usuario al que se le va a compartir el caso	Salida	Se crea la compartición entre los dos usuarios y el caso	
Procedimiento	<ul style="list-style-type: none">• Una vez logueado elegir la opción de compartir de un caso ya simulado• El sistema redirige al panel de compartición• Introducir el email de un usuario existente y que no sea propietario del caso			

	<ul style="list-style-type: none"> El sistema comprueba la validez del email y crea la compartición. La cual se guarda su información en la base de datos.
Precondición	El usuario debe estar logueado y elegir la opción de compartir de un caso ya simulado
Postcondición	El sistema debe crear la nueva compartición
Caso de uso	CU-04_3, CU-05_1

Tabla 267 PN-08

PN-09			
Objetivo	Descargar un caso		Componentes JSP pages, Servlets, JPA, Base de datos
Entrada	Petición del usuario de descargar el caso	Salida	Inicio de la descarga del caso
Procedimiento	<ul style="list-style-type: none"> Una vez logueado elegir la opción de descargar de un caso ya simulado El sistema carga de la base de datos la información del caso y busca el archivo en el servidor. El sistema lanza la descarga del archivo. 		
Precondición	El usuario debe estar logueado y elegir la opción de descargar de un caso ya simulado		
Postcondición	El sistema debe generar la descarga		
Caso de uso	CU-04_4		

Tabla 268 PN-09

PN-10			
Objetivo	Borrar un caso		Componentes JSP pages, Servlets, JPA, Base de datos
Entrada	Petición del usuario de borrar el caso	Salida	Se elimina del servidor el caso y se mueve el registro en la base de datos de ese caso a la tabla deletedProjects
Procedimiento	<ul style="list-style-type: none"> Una vez logueado elegir la opción de borrar de un caso ya simulado o con errores El sistema pide confirmación Si esta es afirmativa el sistema borra del servidor el caso y mueve su registro en la base de datos a la tabla deletedProjects 		
Precondición	El usuario debe estar logueado y elegir la opción de borrar de un caso ya simulado o con errores		
Postcondición	El sistema debe eliminar del sistema el caso y mover su registro en la base de datos a la tabla de casos borrados		
Caso de uso	CU-04_5		

Tabla 269 PN-10

PN-11			
Objetivo	Eliminar una compartición		Componentes JSP pages, Servlets, JPA, Base de datos

Entrada	Petición del usuario de eliminar una compartición con un usuario de un caso	Salida	Se elimina la compartición y el usuario compartido pierde el acceso a ese caso
Procedimiento	<ul style="list-style-type: none"> Una vez logueado y dentro del panel de comparticiones de un caso eliminar la compartición de un usuario que tiene acceso al caso. El sistema borra de la base de datos el registro de esa compartición 		
Precondición	El usuario debe eliminar la compartición con un usuario de un caso que haya compartido		
Postcondición	El sistema debe eliminar esa compartición de los registros		
Caso de uso	CU-05_2		

Tabla 270 PN-11

PN-12			
Objetivo	Modificar permisos de compartición	Componentes	JSP pages, Servlets, JPA, Base de datos
Entrada	Petición del usuario de modificar los permisos de una compartición	Salida	Se actualiza la compartición en la base de datos
Procedimiento	<ul style="list-style-type: none"> El usuario cambia los permisos de una compartición de un caso con un usuario El actualiza la compartición en la base de datos 		
Precondición	El usuario debe modificar los permisos de una compartición existente de un caso		
Postcondición	El sistema debe actualizar los permisos de esta compartición		
Caso de uso	CU-05_3		

Tabla 271 PN-12

PN-13			
Objetivo	Elegir idioma de la aplicación	Componentes	JSP pages, Servlets
Entrada	Petición del usuario de cambio de idioma	Salida	Se actualiza el idioma de la aplicación y se recarga la página actual con el cambio
Procedimiento	<ul style="list-style-type: none"> El usuario elige un idioma distinto del actual El sistema actualiza el idioma elegido en la sesión del usuario El sistema recarga la página actual con el idioma cambiado 		
Precondición	El usuario debe elegir un idioma distinto del actual		
Postcondición	El sistema debe recargar la página actual con el idioma cambiado y mantener este como idioma por defecto del usuario hasta próximo cambio		
Caso de uso	CU-06		

Tabla 272 PN-13

PN-14			
Objetivo	Paginación de casos de simulación	Componentes	JSP pages, Servlets, JPA, Base de datos
Entrada	Petición del usuario de ir a página anterior/siguiente de	Salida	El sistema busca los casos a mostrar en la página anterior/siguiente

	la lista de casos del repositorio		
Procedimiento	<ul style="list-style-type: none"> Una vez logueado elegir la opción página siguiente anterior de la lista de casos El sistema carga en la nueva página los casos que no se mostraban en la anterior página 		
Precondición	El usuario debe tener al menos más de cuatro casos propios o compartidos en la aplicación		
Postcondición	El sistema debe cargar en la lista los casos que antes no estaba mostrando		
Caso de uso	CU-07		

Tabla 273 PN-14

8.3. Análisis de consistencia

En este apartado se comprueba la consistencia de las pruebas especificadas en los apartados anteriores. Para ello toda prueba debe estar relacionada con un requisito.

A continuación se muestra una tabla por cada tipo de pruebas consideradas en la que se muestra los requisitos con los que está relacionado cada una.

Prueba Unitaria	Requisito software relacionado
PU-00	SR-F00
PU-01	SR-F05, SR-F09, SR-F10
PU-02	SR-F04
PU-03	SR-F11
PU-04	SR-F16
PU-05	SR-F12
PU-06	SR-F10
PU-07	SR-F06
PU-08	SR-F07
PU-09	SR-F04
PU-10	SR-F16, SR-F17, SR-F18, SR-F19
PU-11	SR-F11
PU-12	SR-F01, SR-F03
PU-13	SR-F04
PU-14	SR-F16
PU-15	SR-F08
PU-16	SR-F11
PU-17	SR-F20

Tabla 274 PU/SR

Prueba de Aceptación	Requisito de usuario relacionado
PA-00	UR-C00, UR-C04, UR-C06, UR-C07, UR-R11
PA-01	UR-C01
PA-02	UR-C02, UR-R03

PA-03	UR-C09, UR-R05
PA-04	UR-C10, UR-C06
PA-05	UR-C11
PA-06	UR-C12
PA-07	UR-C13
PA-08	UR-C14, UR-C15
PA-09	UR-C16, UR-R10
PA-10	UR-18

Tabla 275 PA/UR

Prueba de Sistema	Requisito software relacionado
PS-00	SR-C00
PS-01	SR-C00
PS-02	SR-F14, SR-R02

Tabla 276 PS/SR

Prueba de Implantación	Requisito software relacionado
PI-00	SR-R00, SR-R01
PI-01	SR-R02
PI-02	SR-S01

Tabla 277 PI/SR

Prueba de integración	Caso de uso
PN-00	CU-00_1
PN-01	CU-00_2
PN-02	CU-00_3
PN-03	CU-01
PN-04	CU-02
PN-05	CU-03
PN-06	CU-04_1
PN-07	CU-04_2
PN-08	CU-04_3, CU-05_1
PN-09	CU-04_4
PN-10	CU-04_5
PN-11	CU-05_2
PN-12	CU-05_3
PN-13	CU-06
PN-14	CU-07

Tabla 278 PN/SR

9. Conclusiones

A continuación se recogen las conclusiones obtenidas tras la finalización del proyecto, también se hará un breve análisis de las posibles mejoras futuras para este proyecto al final de este apartado.

La elaboración del proyecto ha seguido correctamente la metodología de desarrollo Métrica 3, por lo que este documento contiene las principales fases que recoge esta metodología sobre el desarrollo de proyectos software: requisitos, análisis, diseño, tecnologías de desarrollo, estimaciones y gestión,...

El diseño completo realizado en este documento ha servido de manera satisfactoria a la posterior codificación, ya que ha agilizado bastante este proceso al tener las bases de lo que se quería hacer.

Respecto a la elección de Java Servlets en lugar de PHP para la arquitectura del proyecto se cree que ha sido la elección correcta para lo que se buscaba y la que mejor se adaptaba al metapatrón elegido de MVC+Front Controller. Además, en vistas a futuras mejoras y extensiones, Java Servlets es una alternativa más escalable que PHP.

La estimación realizada para este proyecto ha sido la adecuada y se han podido respetar los inicios y finales de las tareas que componen este proyecto. Y sobre todo, lo más importante, se ha respetado la fecha de entrega final.

Todos los requisitos han sido cumplidos y se ha verificado este dato en las diversas matrices y análisis de consistencia que se han realizado en los diferentes puntos de este documento.

En general, la realización del proyecto ha sido completa y satisfactoria en cuanto a los requerimientos especificados y al tiempo dado.

9.1. Posibles mejoras

Pese a que esta primera versión de la aplicación web es bastante completa se pueden considerar las posibles mejoras y extensiones de funcionalidad para el futuro:

- **Nuevos idiomas:** Aunque la aplicación inicial ya recoge el español e inglés nunca está de más añadir idiomas importantes para llegar al máximo de usuarios posibles.
- **Mejora de accesibilidad:** Aunque se ha respetado este punto en cuanto a la elección de colores y tamaños de fuente e iconos en la aplicación se puede mejorar para hacerla lo más accesible a todo tipo de personas.
- **Diseño responsive:** Quizás una de las más importante. La aplicación tendría un mayor uso si el diseño de esta permitiera su uso de manera eficiente desde diferentes dispositivos electrónicos (smartphones, tablets,...) con resoluciones dispares.
- **Compatibilidad con diferentes tipos de proyectos/casos:** Para esta primera versión se ha elegido un tipo específico de casos que pueden ser simulados en un servidor. Podría mejorarse este aspecto introduciendo a la aplicación compatibilidad con

otros tipos de casos/proyectos que pudieran ser simulados en sus servidores de simulaciones respectivos. Para ello habría que introducir la pequeña funcionalidad para establecer el Web Services entre los diferentes servidores de simulaciones y la aplicación web.

- **Máximo de espacio usado por usuario:** No se ha tenido en consideración para esta primera versión el espacio total ocupado por los casos subidos en el servidor de cada usuario. Para próximas versiones sería adecuado poner una cuota máxima con el fin de no perder rendimiento en la aplicación o evitar posibles ataques intentando colapsar la capacidad de almacenamiento del servidor.

10. Bibliografía

A continuación se exponen todo el material que ha servido de apoyo para la elaboración de este documento:

- [1]. E.Muckenhuber y J.Morgan, *DataSource configuration*, 2013. Disponible en: <https://docs.jboss.org/author/display/WFLY8/DataSource+configuration>
- [2]. S.Marlow y G.Turner, *JPA Reference Guide*, 2014. Disponible en: <https://docs.jboss.org/author/display/WFLY8/JPA+Reference+Guide#JPAReferenceGuide-BindingEntityManagerFactory%2FEntityManagertoJNDI>
- [3]. Stack Overflow [en línea] <http://stackoverflow.com/>
- [4]. S.Lozano, "Análisis, diseño y desarrollo de un sistema web de gestión de encuestas", 2013, Director: I. González. Proyecto Fin de Carrera Universidad Carlos III. Disponible en: <http://hdl.handle.net/10016/18498>
- [5]. MÉTRICA VERSIÓN 3. Metodología de Planificación, Desarrollo y Mantenimiento de sistemas de información.
- [6]. bitriding.com, *Validar documentos XML usando XSD*, 2013. Disponible en: <http://www.bitriding.com/articulos/validacion-con-xsd.html>
- [7]. S.Mitroff, "OneDrive, Dropbox, Google Drive and Box: Which cloud storage service is right for you?", C|net, 2014, Disponible en: <http://www.cnet.com/how-to/onedrive-dropbox-google-drive-and-box-which-cloud-storage-service-is-right-for-you/>
- [8]. Mkyong, *JavaMail API – Sending email via Gmail SMTP example*, 2012, Disponible en: <http://www.mkyong.com/java/javamail-api-sending-email-via-gmail-smtp-example/>
- [9]. W.Crawford y J.Kaplan, *J2EE Design Pattern: FrontController pattern Java code example*, 2003. Disponible en: <http://www.javaqaq.nu/java-example-code-1033.html>
- [10]. Tecnologías informáticas para la web. Universidad Carlos III de Madrid.
- [11]. M. Meno, *Dropzone*, 2012, Disponible en: <http://www.dropzonejs.com/>

ANEXO I. Manual de usuario

El presente manual sirve de orientación al usuario sobre el manejo de la aplicación. En este manual se hará una descripción y se detallarán los pasos a seguir sobre el uso de las funcionalidades del programa.

1. Acceso a la aplicación

Mediante un explorador web el usuario debe acceder a la URL de la página de inicio de la aplicación.

Esta página inicial mostrará un acceso por login, el cual pedirá el email y contraseña de registro para acceder al repositorio.

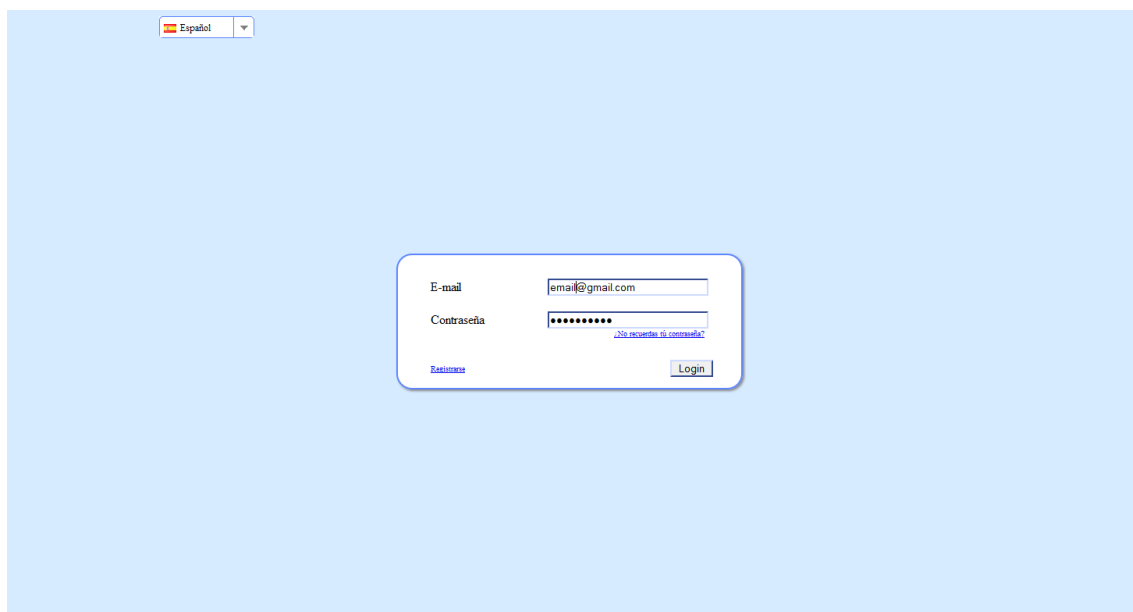
The image shows a web browser window with a light blue background. At the top left, there is a language dropdown menu set to 'Español'. In the center, there is a white login form with a blue border. The form contains two input fields: 'E-mail' with the text 'email@gmail.com' and 'Contraseña' with masked characters '*****'. To the right of the password field is a blue link that says '¿No recuerda su contraseña?'. Below the password field is a blue link that says 'Registrarse'. At the bottom right of the form is a 'Login' button.

Ilustración 129 Pantalla login

Si el usuario ya dispone de una cuenta deberá introducir los datos correctamente y pulsar 'Login'. Si los datos son correctos se redirigirá a la pantalla del repositorio, en cambio si estos son erróneos se notificará de ello mediante el siguiente error.

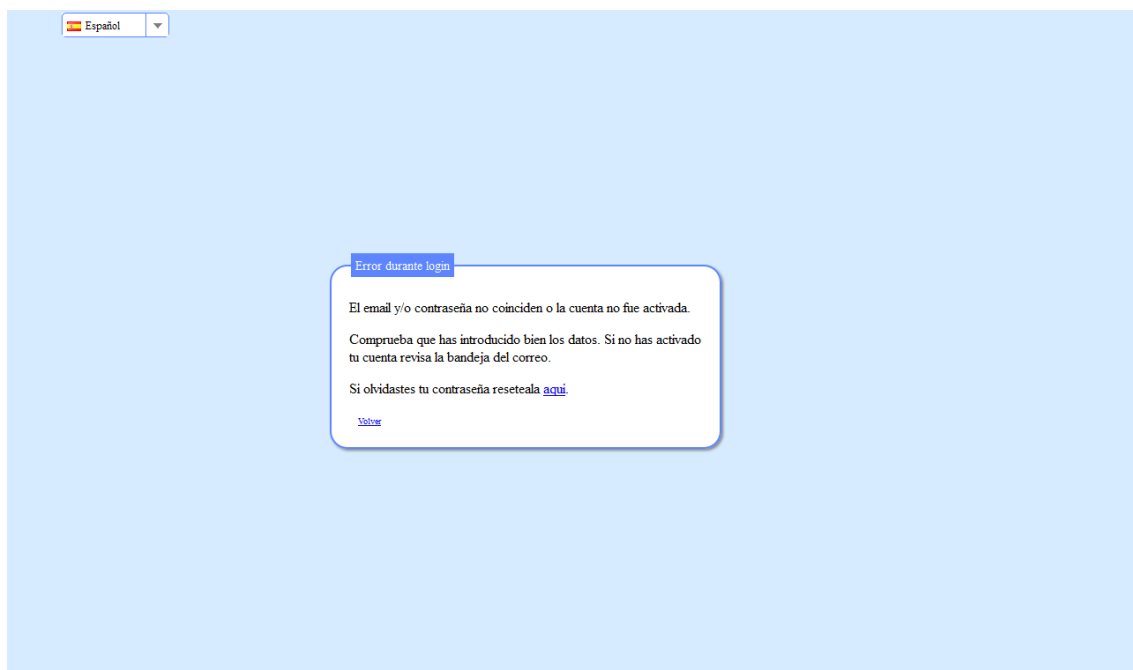


Ilustración 130 Error login

Si el usuario no dispone de una cuenta podrá crear una nueva usando el link llamado ‘*Registrarse*’ que hay en la página de login. Este link le redirigirá a un formulario donde deberá introducir los datos de su nueva cuenta. Posteriormente, tras crearla, deberá activarla con el link del correo que le llegará a su email de registro.

A screenshot of a web application interface with a light blue background. In the top left corner, there is a language selector showing 'Español' with a dropdown arrow. In the center, there is a white rounded rectangle with a blue border. At the top of this rectangle is a blue header with the text 'Formulario de registro'. Below the header, there are four input fields: 'E-mail' with the value 'email@gmail.com', 'Nombre' with the value 'Nombre', 'Apellidos' with the value 'Apellido1 Apellido2', and 'Contraseña' with the value '*****'. Each input field has a small asterisk icon to its right. Below the input fields, there is a blue link 'Volver' on the left and a blue button 'Registrar' on the right.

Ilustración 131 Panel de registro

Si por el contrario, dispone de una cuenta pero no se acuerda de la contraseña use el link llamado ‘*¿No recuerdas tu contraseña?*’ donde deberá introducir el correo de registro. A este le llegará un email con un link que deberá visitar para cambiar la contraseña de su cuenta.



The screenshot shows a web interface with a light blue background. In the top left corner, there is a language selector showing 'Español' with a dropdown arrow. Centered on the page is a white rounded rectangle with a blue border. At the top of this rectangle is a blue header with the text 'Recuperación de contraseña'. Below the header, there is a paragraph of text: 'Por favor, introduce tu dirección de e-mail. En breve, recibirás un mensaje con las instrucciones necesarias para restablecer tu contraseña.' Underneath this text is a label 'E-mail' followed by a text input field. Below the input field is a blue link that says 'Volver'. To the right of the 'Volver' link is a blue button with the text 'Restablecer contraseña'.

Ilustración 132 Recuperar contraseña

Si tras seguir estos pasos no consigues ingresar a la aplicación póngase en contacto con la persona encargada de gestionar la aplicación en su equipo de trabajo o empresa.

2. Cambiar datos de registro

Una vez ingresado satisfactoriamente en su repositorio podrá cambiar los datos de registro de su cuenta (contraseña, nombre y apellidos). Para ello elija la opción del menú superior 'Cuenta'.

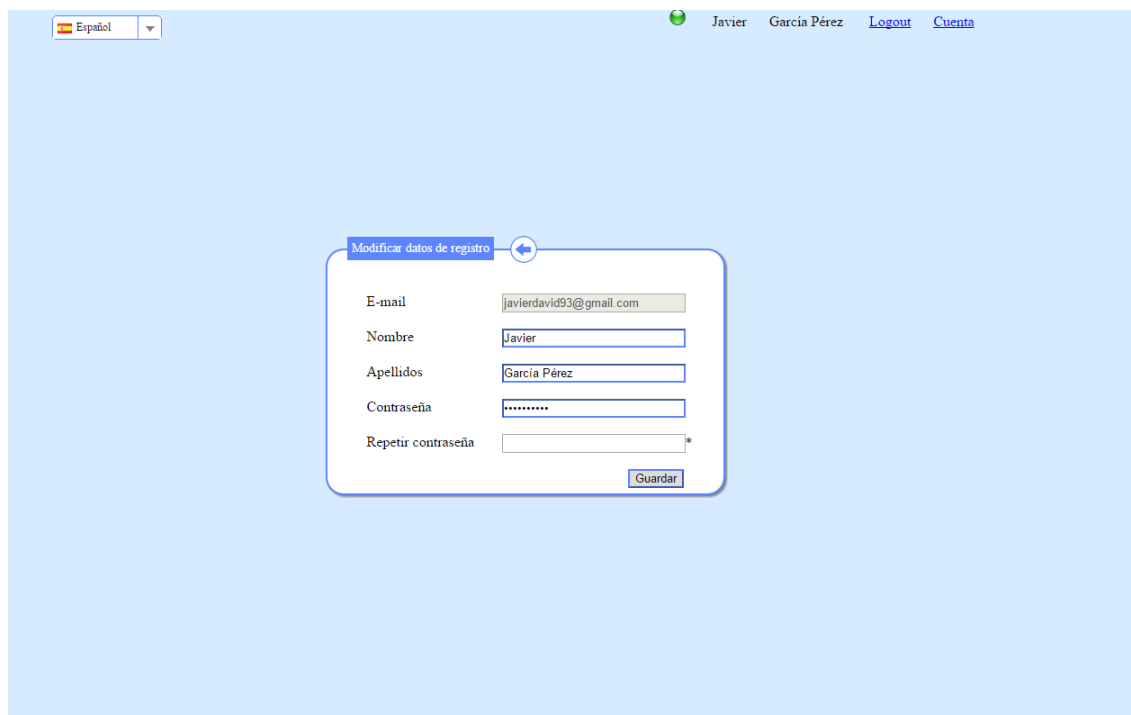



Ilustración 133 Datos de registro

Por motivos de seguridad no se le permitirá cambiar el email. Si este cambio es una necesidad importante para usted póngase en contacto con la persona encargada de la aplicación en su entorno.

3. Subir caso de simulación

En la pantalla de repositorio haga clic en el icono  (izquierda del título de la ventana). Será redirigido a una nueva vista donde podrá subir a la aplicación los casos que disponga de una copia local en su equipo.

Antes de subir un caso o de simularlo verifique que el servidor de simulaciones esté operativo. El LED del menú superior será verde cuando todo sea correcto y será rojo cuando haya algún problema. Pasando el puntero de ratón por encima de él podrá acceder a más información. El sistema por seguridad desactivará las subidas y simulaciones de casos cuando el servidor de simulaciones no esté disponible.

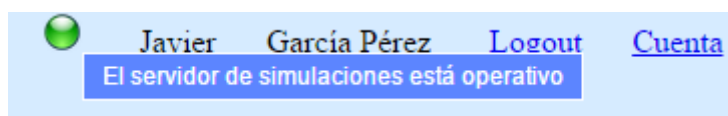


Ilustración 134 Estado del servidor de simulaciones

Para subir los casos arrástrelos a la caja con línea discontinua o simplemente haga clic en ella. Tras completarse la subida de todos los casos que haya puesto en cola será redirigido al repositorio donde podrá seguir su simulación.

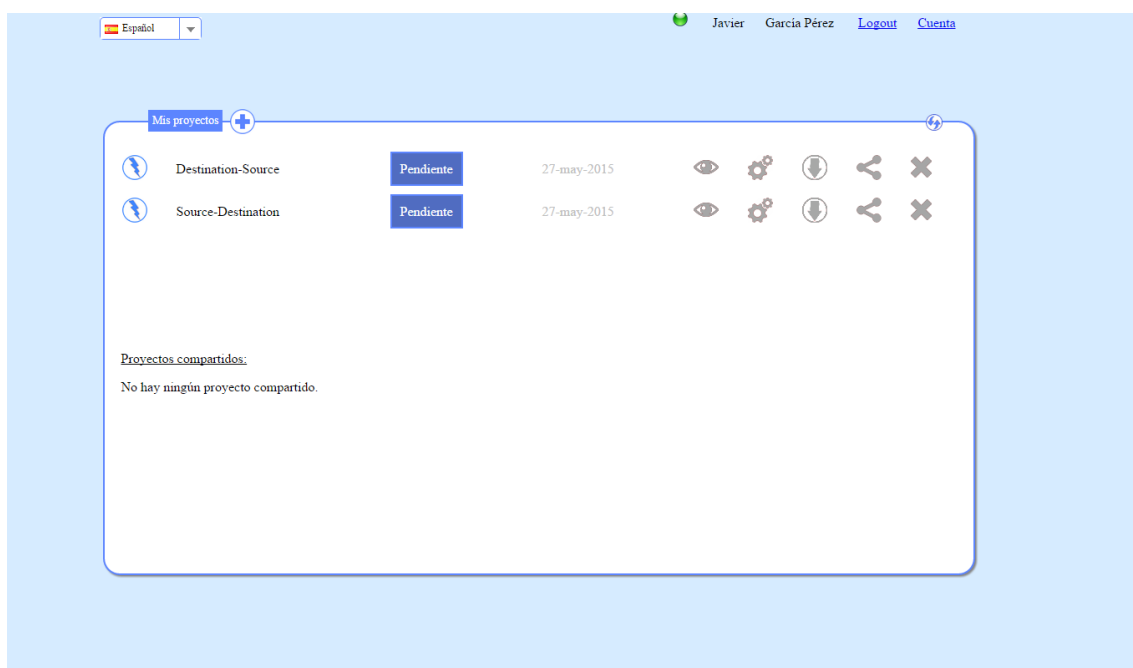
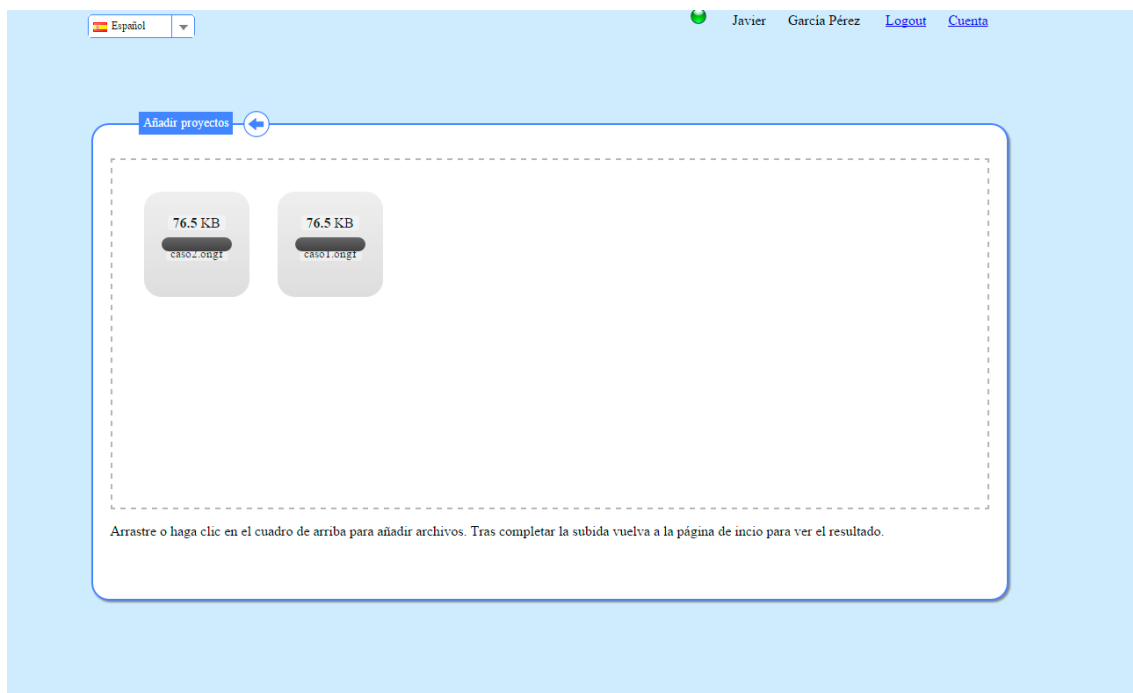


Ilustración 135 Estado simulación

Usted podrá subir casos repetidos, el sistema simplemente ante este caso le notificará por si quiere mantener la copia más reciente o si simplemente quiere mantener todas.

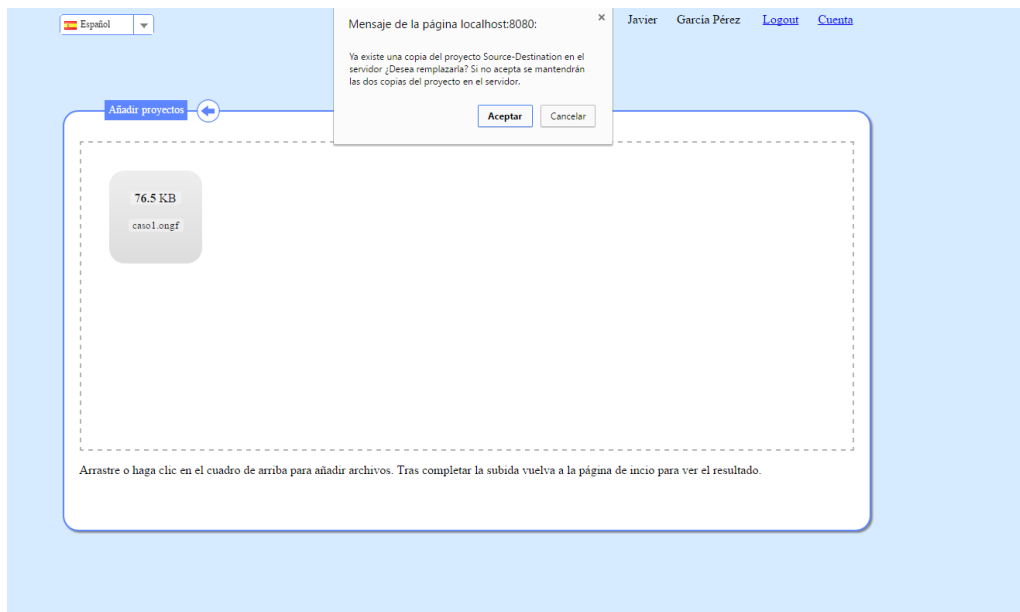


Ilustración 136 Subida casos repetidos

4. Uso del repositorio

Tras un login correcto la pantalla que se le mostrará a continuación será el repositorio donde podrá gestionar todos los casos que suba.

Por cada caso subido y simulado se le habilitarán una serie de opciones que podrá realizar sobre él. Los iconos que aparezcan de en azul serán las opciones habilitadas, mientras los que aparezcan en gris serán a las que no tiene acceso.

Estas opciones que puede usar sobre los casos serán la de ver su solución (👁️), re-simular (⚙️), descargar (⬇️), compartir (🔗) y eliminar (✖️).

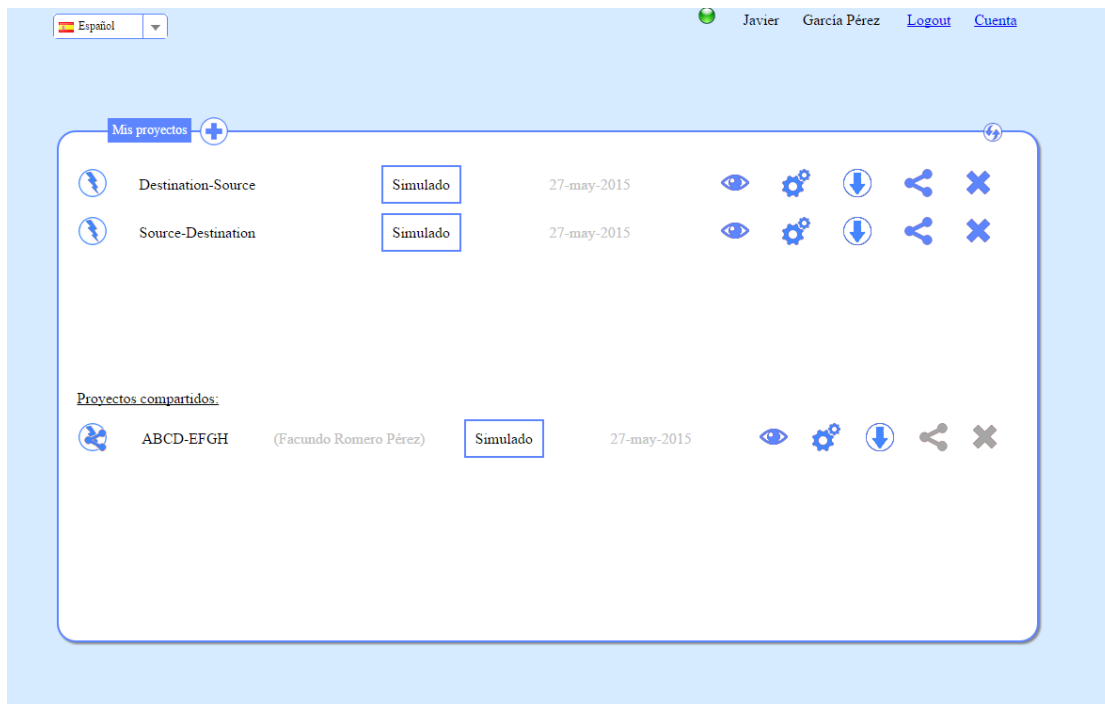


Ilustración 137 Vista del repositorio

Si elige la opción de ver solución (👁️) será redirigido a una vista dentro de la aplicación donde podrá ver el contenido de esta. El contenido de la solución dependerá del servidor de simulaciones donde se simuló el proyecto por lo que la aplicación no es responsable del contenido mostrado.

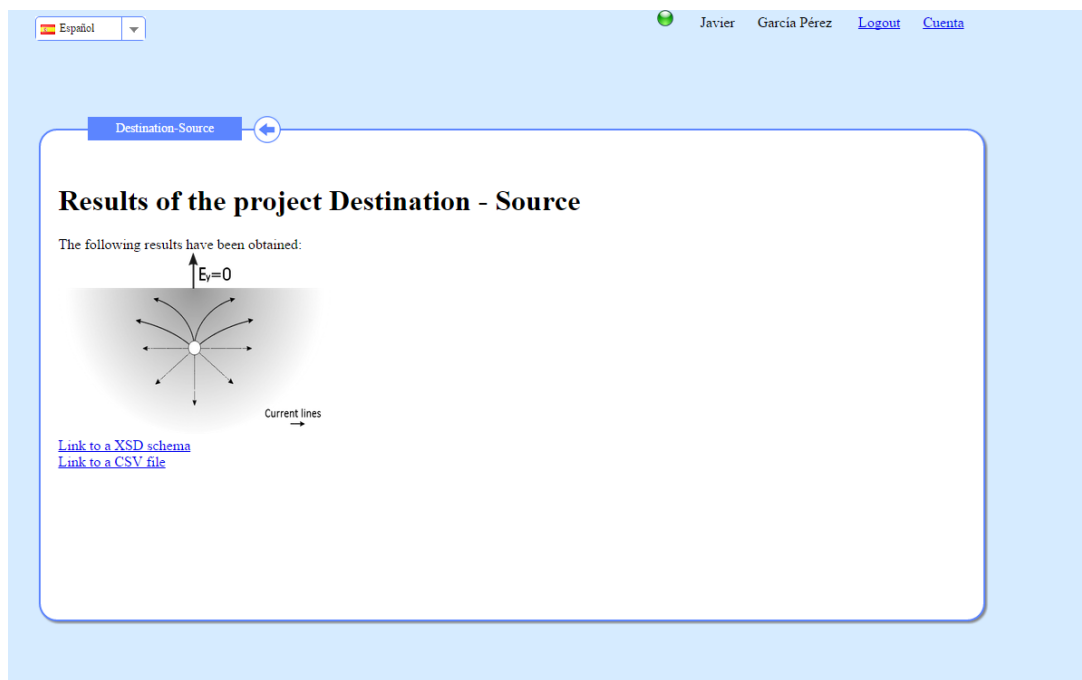


Ilustración 138 Ver solución caso

La opción de re-simular (⚙️) volverá a lanzar este proceso por si la solución no es correcta o hubo un error durante el proceso.

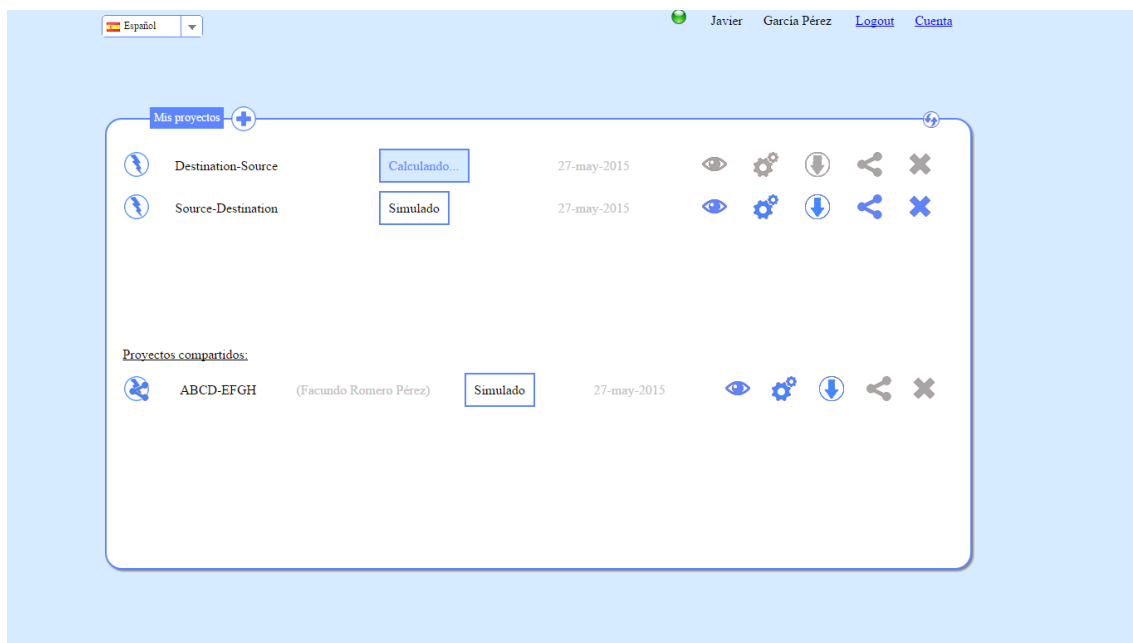


Ilustración 139 Re-simular caso

La función descargar (📥) bajará a su equipo una copia comprimida del caso ya simulado.

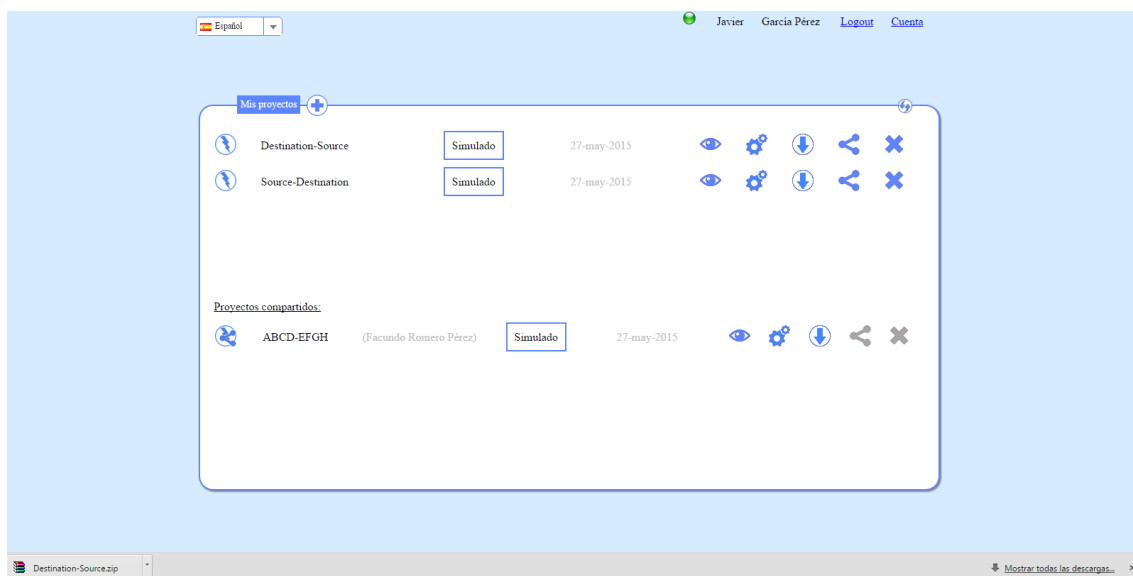


Ilustración 140 Descargar caso

El icono compartir (🔗) permitirá dar acceso a otros usuarios de la aplicación que no sean sus propietarios. Esta funcionalidad al ser más compleja se explicará en el apartado 5. *Compartir caso*.

La opción de borrar un caso (✖) quitará de su repositorio el proyecto y de los repositorios de los usuarios que tengan acceso a él debido a una compartición del mismo. Para mayor seguridad se le pedirá confirmación a la hora de borrar un caso.

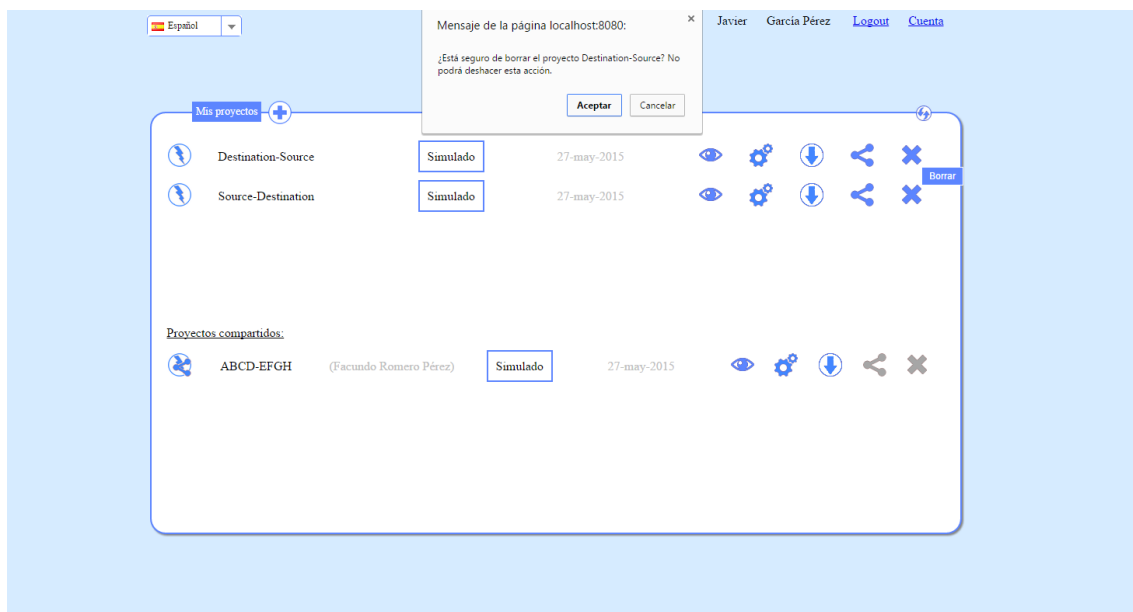


Ilustración 141 Borrar un caso

5. Compartir caso

Para realizar esta acción sobre un caso ya simulado seleccione el icono compartir (🔗). Será redirigido a una página donde deberá introducir en el primer cuadro de texto el email del usuario al que desea dar acceso al caso, posteriormente pulse el botón añadir (+). Para facilitar la búsqueda usted podrá introducir el nombre y apellidos del usuario y el sistema le sugerirá el email.

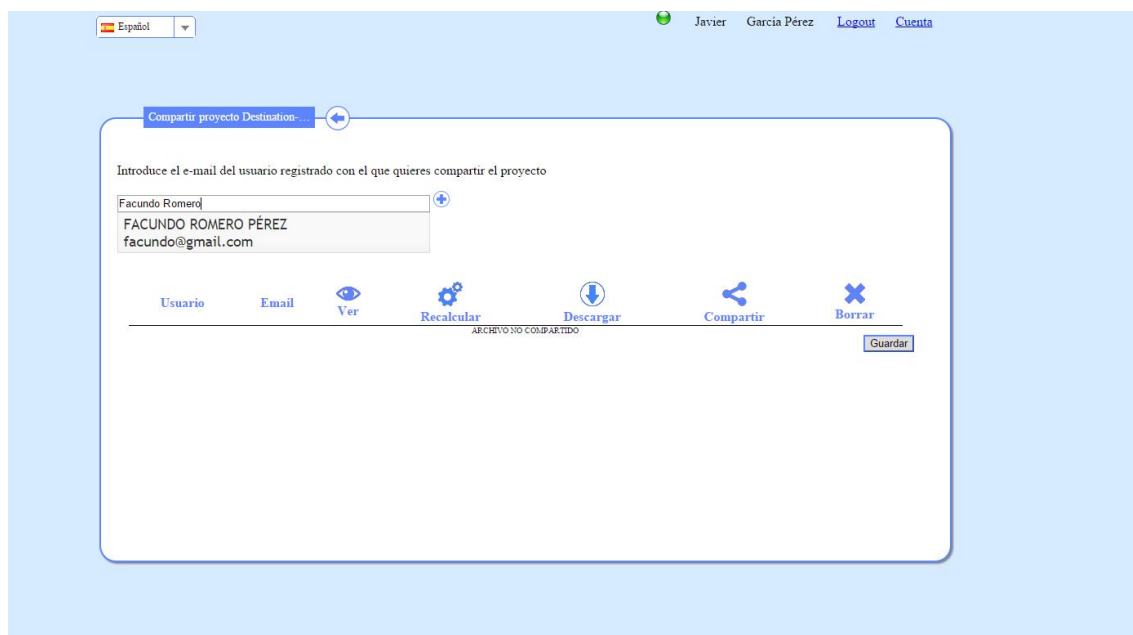


Ilustración 142 Compartir caso (1)

Una vez añadido un usuario podrá gestionar sus permisos de gestión del caso en la tabla inferior marcando las cajitas que vea necesarias y pulsando posteriormente el botón de

‘Guardar’. Para eliminar la compartición de un usuario con un caso simplemente pulse el icono eliminar (🗑️) a la derecha de la fila de permisos del usuario en cuestión.

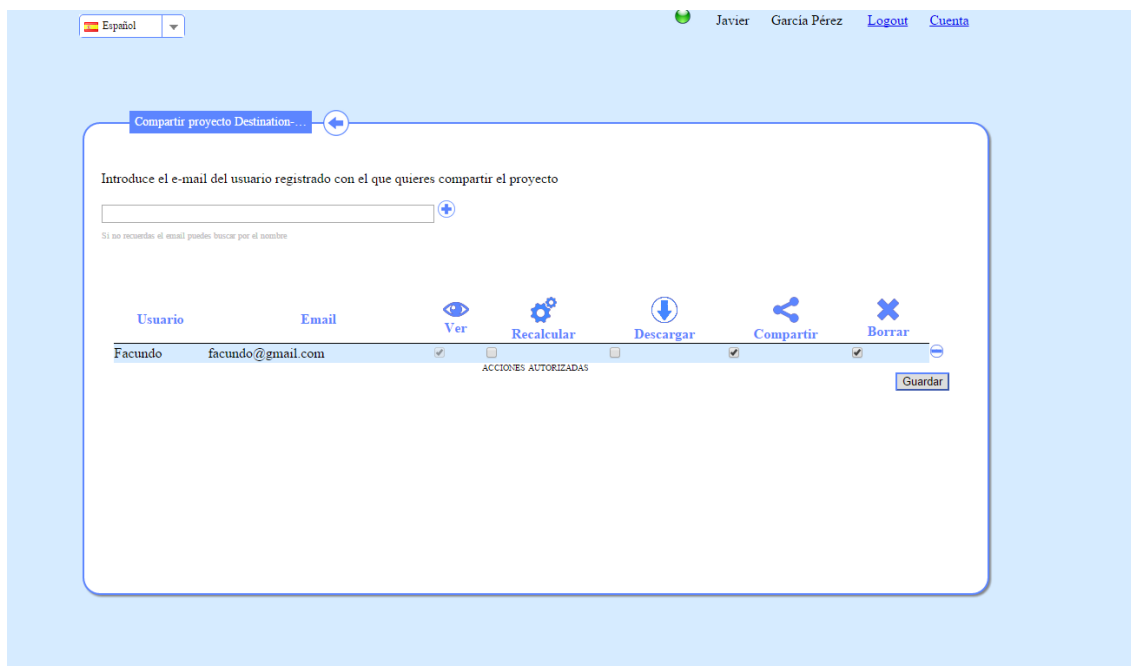


Ilustración 143 Compartir caso (2)

Las acciones que hagan los usuarios sobre los casos a los que proporcionó acceso es responsabilidad suya, por lo que se recomienda que tenga precaución a la hora de compartir y asignar permisos sobre los proyectos.

6. Desconectar de la aplicación

La aplicación por defecto cerrará su sesión si esta pasa un total de treinta minutos inactiva. Para cerrarla de forma manual use el link ‘Logout’ del menú superior.

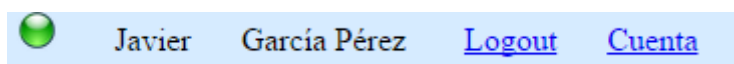


Ilustración 144 Cerrar sesión

Se recomienda cerrar manualmente las sesiones y no esperar que lo haga la propia aplicación para no sobrecargarla innecesariamente.

7. Cambiar idioma

La aplicación cuenta con soporte para los idiomas inglés y español. Usted puede elegir el idioma deseado mediante el selector que se encuentra en la parte superior izquierda de la aplicación.

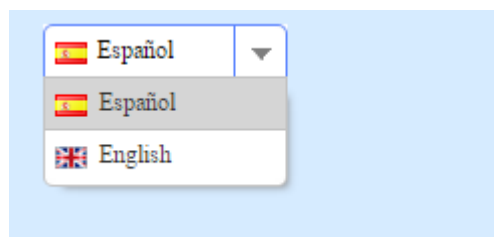


Ilustración 145 Cambio de idioma

El cambio de idioma estará disponible en cualquier página de la aplicación web.

8. Navegación entre listado de casos

Puede darse el caso en el que cuente con una cantidad considerable de casos subidos. La aplicación web no podrá mostrar la información todos ellos en una única vista, para ello se le habilitará la opción de cambiar de página de la vista de repositorio. De esta manera accederá a la gestión de todos los casos subidos o que tenga acceso por medio de compartición.

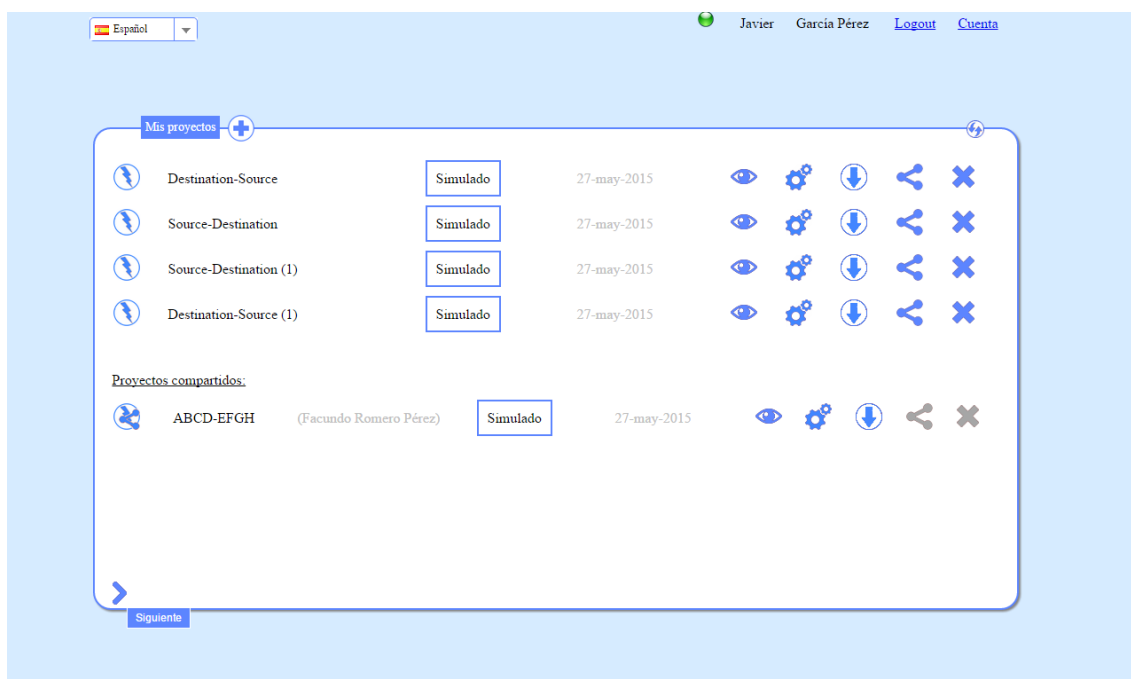


Ilustración 146 Página siguiente/anterior (1)

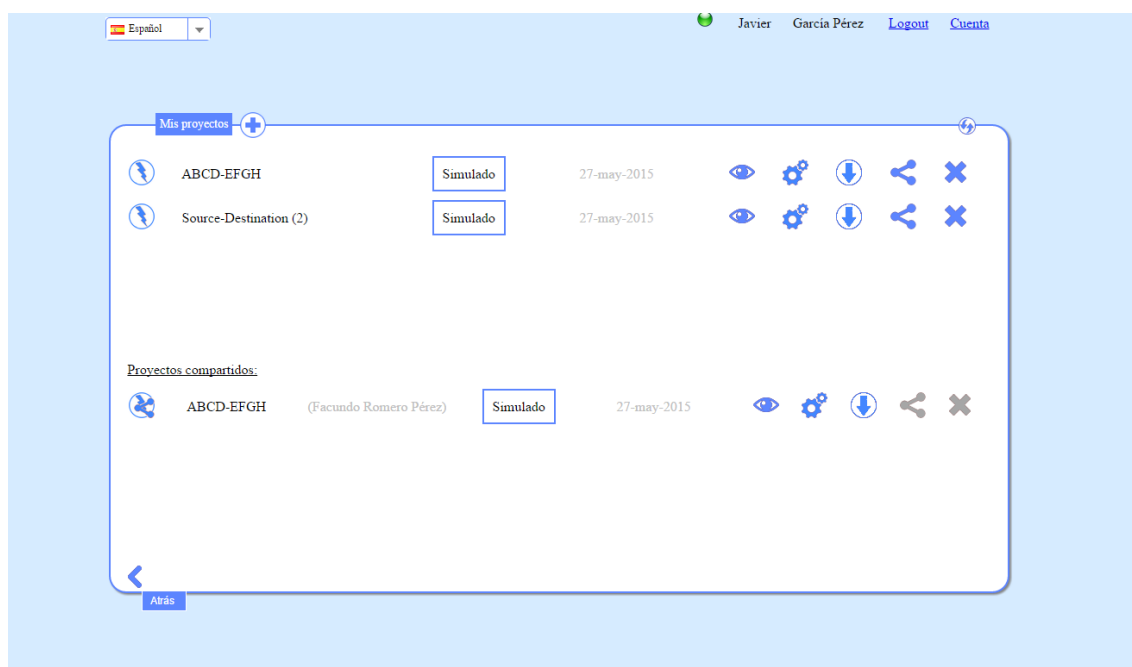


Ilustración 147 Página siguiente/anterior (2)

El cambio de página aparecerá en el caso de que cuente con más de cuatro casos propios o más de cuatro comparticiones.

ANEXO II. Manual de instalación

Este anexo explicará cómo hacer una instalación local de la aplicación para realizar todo tipo de pruebas que se deseen.

1. Base de datos

Descargue y configure en su equipo MySQL en el caso de que no lo tenga instalado. Una vez completado este proceso importe la base de datos del archivo *tfg.sql* que se le entregó en la documentación.

Este archivo contiene la estructura indispensable para poder almacenar la información que posteriormente se guarde y gestione en la aplicación web.

2. Servidor de aplicaciones

El servidor de aplicaciones que deberá instalar es Wildfly en su versión 8.2. Disponible en el siguiente link <http://wildfly.org/downloads/>

Se recomienda, una vez descargado y extraído, realizar dos copias de la carpeta, ya que una de ellas servirá para ejecutar un servidor de simulaciones de prueba.

En la carpeta que usará para desplegar la aplicación web diríjase a la ruta *standalone\deployments* y pegue el archivo *mysql-connector-java-5.1.27-bin.jar* que le fue entregado con la documentación.

Pruebe a lanzar el servidor desde una ventana de comandos con la siguiente instrucción *standalone.bat -c standalone-full.xml* para ello deberá colocarse en la carpeta *bin* del directorio donde extrajo el contenido del servidor.

Con el servidor lanzado diríjase al panel de administración de este entrando a su navegador web y poniendo la siguiente dirección <http://localhost:8080/>, después elija la opción de *Administration Console*. Al ser la primera vez que entra le pedirá que cree una cuenta. Siga los pasos que le indican y cree los dos tipos de cuenta. La cuenta de la opción *b)* deberá tener los siguientes datos:

- **user:** user
- **pass:** password
- **rol:** guest

Una vez creadas reinicie el servidor y vuelva a entrar al panel de administración ahora introduciendo los datos que uso para crear la cuenta de tipo *a)*. Diríjase a *Configuration/Connector/Datasources[1]*. Deshabilite o borre si ya existe un datasource creado y dé en la opción *Add*. En el primer paso introduzca los siguientes datos:

- **Name:** aplicacionDS
- **JNDI:** java:jboss/jdbc/aplicación

En el paso dos elija como JDBC driver el introducido posteriormente en la carpeta *deployments* (mysql-connector-java-5.1.27-bin.jar).

Finalmente en el paso tres introduzca los datos de conexión a la base de datos, estos deberían ser:

- **Connection URL:** jdbc:mysql://localhost:3306/<Nombre dado a la base de datos, en su defecto *tfg*>
- **Username:** usuario con permisos de conexión, consulta, inserción, actualización y borrado sobre esa base de datos. Si no tiene ninguno creado use mismamente al usuario *root*.
- **Password:** contraseña del usuario
- **Security Domian:** déjelo en blanco.

Una vez creado el datasource en la parte de abajo elija la pestaña *Properties* he introduzca la siguiente:

- **Name:** JtaManaged
- **Value:** false

Asegúrese que el datasource está habilitado en su panel tras estos últimos cambios. Apague el servidor y vaya a la carpeta *standalone/configuration* de su servidor y abra el archivo *standalone-full.xml* con un editor de texto, preferiblemente notepad++. Una vez abierto busque la etiqueta *default-bindings* y localice la propiedad *datasource* esta debería ser igual a *java:jboss/jdbc/aplicación*, si no es así cámbielo.

Dentro del mismo documento busque la etiqueta *datasources* y localice el creado recién. Debajo de la línea *<driver>mysql-connector-java-5.1.27-bin.jar</driver>* pegue el siguiente código:

```
<pool>

    <min-pool-size>10</min-pool-size>

    <max-pool-size>500</max-pool-size>

    <prefill>true</prefill>

</pool>
```

Tras estos pasos ya tiene configurado el servidor que desplegará la aplicación web.

3. Servidor de simulaciones

En la carpeta copia que hizo del servidor abra el archivo *standalone-full.xml* con un editor de textos y localice la etiqueta *socket-binding-group*, final del documento, cambie el valor de la propiedad *port-offset* por *\${jboss.socket.binding.port-offset:100}*. De esta manera nuestro servidor de simulaciones de testeo no pisará el puerto de el de la aplicación.

Después realice los pasos de los párrafos 4 y 5 del apartado anterior sobre este servidor. Lance el servidor con la instrucción *standalone.bat -c standalone-full.xml* dentro de la carpeta *bin* y compruebe que no hay ningún fallo en su lanzamiento.

4. Montar aplicación

Apague los dos servidores. Diríjase a la carpeta *standalone\deployments* de cada uno. En el servidor que ha configurado para que soporte la aplicación web pegue el archivo *aplicacionEAR.ear* que se la ha proporcionado con la documentación, haga lo mismo en el otro servidor pegando el archivo *simulacionEAR.ear*.

Inicie los dos servidores con el comando *standalone.bat -c standalone-full.xml*. Una vez finalizada la ejecución abra su navegador y acceda a <http://localhost:8080/aplicacion/>. Debería ver la siguiente pantalla.

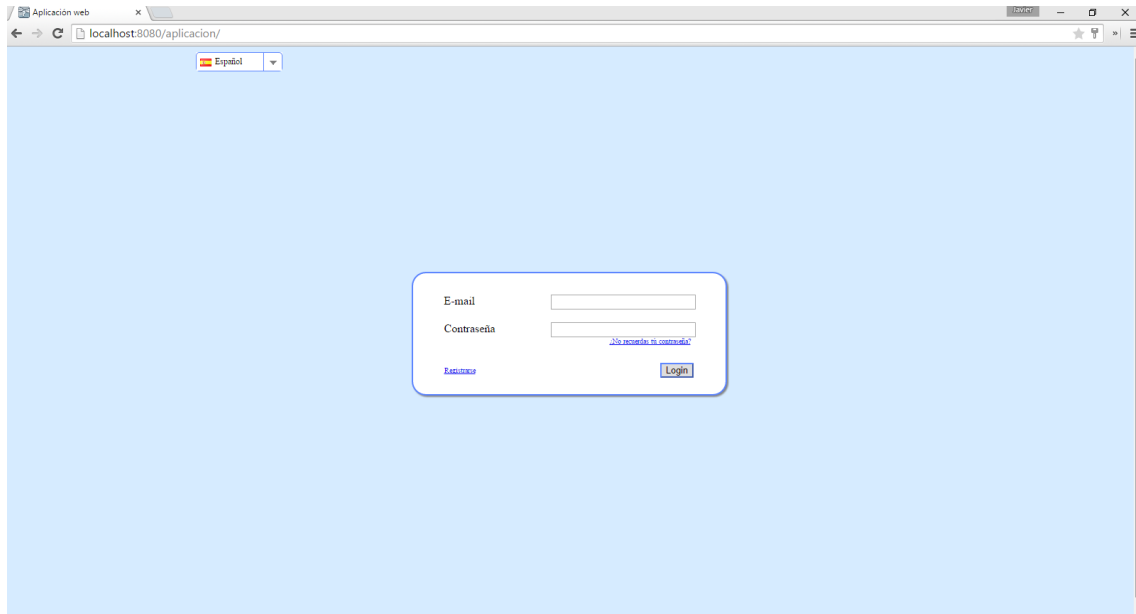


Ilustración 148 Instalación